University of Dundee

University of Dundee

DOCTOR OF PHILOSOPHY

A Quantitative Exploration of Causes of False Positive Single Nucleotide Polymorphisms in Next-Generation Sequencing Data

Bello Ribeiro, Antonio Claudio

Award date:
2016

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 05. May. 2023

A QUANTITATIVE EXPLORATION OF CAUSES OF FALSE POSITIVE SINGLE NUCLEOTIDE POLYMORPHISMS IN NEXT-GENERATION SEQUENCING DATA

By

Antonio Claudio Bello Ribeiro

A THESIS SUBMITTED FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

AT

UNIVERSITY OF DUNDEE

DUNDEE, UNITED KINGDOM

AUGUST 2016

© Copyright by Antonio Claudio Bello Ribeiro, 2016

Contents

List of Tables **vi**

List of Figures **viii**

Acknowledgements **xii**

Abstract **xvii**

List of Abbreviations **xix**

1 Literature Review **1**
- 1.1 Introduction . . . 3
- 1.2 Main aspects of the traditional NGS-based approach to SNP discovery 12
  - 1.2.1 Sequencing . . . 13
  - 1.2.2 Variant calling pipeline – Base calling and quality control stage . . . 25
  - 1.2.3 Variant calling pipeline – *De novo* assembly stage for non-model organisms . . . 28
  - 1.2.4 Variant calling pipeline – Alignment stage . . . 39
  - 1.2.5 Variant calling pipeline – Post-alignment stage . . . 44
  - 1.2.6 Variant calling pipeline – SNP calling stage . . . 46
- 1.3 FP SNP examples . . . 52
  - 1.3.1 FP SNPs due to sequencing errors and sequence-specific errors 53
  - 1.3.2 FP SNPs due to duplicates . . . 55
  - 1.3.3 FP SNPs due to reference misassembly . . . 56
  - 1.3.4 FP SNPs due to read mismapping . . . 57

2 False positive SNP generation due to reference misassembly **62**
- 2.1 Introduction . . . 62
- 2.2 Testing for paralogs: an experiment with real RNA-Seq data from the barley cultivar Bowman . . . 70
  - 2.2.1 Methods . . . 70

2.2.2 Results . . . 77
2.3 Testing the misassembly with a *de novo* genome assembler . . . 83
2.3.1 Methods . . . 83
2.3.2 Results . . . 88
2.4 Discussion . . . 101
2.5 Conclusions . . . 110

3 False positive SNP generation due to read mismapping 111
3.1 Introduction . . . 111
3.2 Methods . . . 121
3.2.1 Datasets used . . . 121
3.2.2 Software implementation and use . . . 123
3.3 Results . . . 127
3.4 Discussion . . . 131
3.5 Conclusions . . . 136

4 A multifactorial experiment to evaluate false positive SNP generation due to read mismapping 137
4.1 Introduction . . . 138
4.2 Methods . . . 150
4.2.1 Read datasets preparation . . . 150
4.2.2 Reference genome assembly . . . 150
4.2.3 Read mapping . . . 152
4.2.4 SNP calling . . . 153
4.2.5 Control dataset . . . 155
4.2.6 Read mismapping quantification stage . . . 156
4.2.7 SNP annotation . . . 157
4.2.8 Replicate workflow runs . . . 158
4.2.9 Statistical analysis . . . 159
4.3 Results . . . 160
4.3.1 General observations . . . 160
4.3.2 Main effects and interactions among experimental factors . . . 162
4.3.3 Read mismapping statistics, SNP annotation, and genomic distribution of FP SNP sites . . . 171
4.4 Discussion . . . 173
4.4.1 Role of the reference sequence in the generation of FP SNPs 175
4.4.2 Choice of tools for assembly, mapping, and variant calling and their influence on the generation of FP SNPs . . . 177
4.4.3 The impact of SNP filtering on FP SNP numbers . . . 179
4.4.4 The impact of read length on FP SNP numbers . . . 179
4.4.5 Genomic patterns of FP SNP locations . . . 184

4.4.6 Taking false negative SNPs into consideration 185
4.5 Conclusions 194

5 General conclusions and future work **198**
5.1 General conclusions 198
5.2 Future work 200

Bibliography **205**

Appendix A **234**
A.1 False positive SNP generation due to reference misassembly – supplementary information 234
A.1.1 Commands, parameters, and some detailed results of the experiment with real RNA-Seq data from barley cultivar Bowman to test for reference misassembly 234
A.1.2 Commands, parameters, and some detailed results of the *de novo* assembly experiment with simulated reads from *Arabidopsis thaliana* to test for reference misassembly 237
A.1.3 Software availability 260

Appendix B **261**
B.1 False positive SNP generation due to read mismapping – supplementary information 261
B.1.1 Commands, parameters, and some detailed results of the *de novo* assembly experiment with simulated reads from *Arabidopsis thaliana* to test for read mismapping 261
B.1.2 FP SNP sites genomic locations tables 269
B.1.3 Software availability 269

Appendix C **270**
C.1 A multifactorial experiment to evaluate false positive SNP generation due to read mismapping – supplementary information 270
C.1.1 Read simulation – additional information 270
C.1.2 *De novo* assembly – additional information 278
C.1.3 Read mapping – additional information 286
C.1.4 SNP calling – additional information 291
C.1.5 Pipeline usage in the multifactorial experiment to evaluate the FP SNP generation due to read mismapping 297
C.1.6 SNP manifests extraction 300
C.1.7 SNP annotation detailed results 300
C.1.8 FP SNP sites genomic locations by chromosomes 305

C.1.9 Software availability 308
C.1.10 Supplementary files 308

Appendix D **309**
D.1 List of posters, presentations, and publications 309
D.1.1 International peer-refereed publication 309
D.1.2 Posters and presentations 309

List of Tables

1.1 Comparison of Sanger sequencing and NGS technologies. 25
1.2 Examples of Phred quality scores. 26

2.1 Summary of results from the four different scenarios of mismatches allowed (n = 5, 10, 20, and 30) for the BWA mappings (for the barley dataset) after the run of the pipeline tool with the BLAST search feature turned ON. 78
2.2 Comparison of nine different runs of the pipeline using the same mismatch setting, but varying numbers of the randomly chosen FLcDNAs in the BLAST target database. 82
2.3 Comparison of twelve different runs of the pipeline using the same mismatch setting, but varying numbers of chosen FLcDNAs in the BLAST target database in a deterministic manner. 82
2.4 Summary of results from the four different scenarios of mismatches allowed (n = 5, 10, 20, and 30) for the Bowtie2 mappings (for the *A. thaliana* dataset) after the run of the pipeline tool with the BLAST search feature turned ON. 89
2.5 Five cases of *A. thaliana* annotation BLASTDB hits found by the pipeline for the cases with either the reference or the alternate alleles present in the reads sets considering the most relaxed mapping scenario. 100
2.6 Summary of BLAST database hit counts for the *A. thaliana* paralogy test in the most relaxed mapping scenario. 101

3.1 Number of occurrences retrieved by the used annotation approach. . 128

4.1 Main effects from the factorial Analysis of Variance (ANOVA). . . . 162
4.2 First major higher-order interaction. 165
4.3 Second major higher-order interaction. 166
4.4 Mapper and mapping stringency interaction. 167
4.5 MAPQ and variant caller interaction. 168
4.6 Assembly type, mapper, and mapping stringency interaction. 168

4.7 Assembly and depth filter interaction. 171
4.8 Zero-FP SNP combinations related to the control genome which were selected for the FN SNP experiment. 186
4.9 Final SNP numbers obtained in the FN SNP experiment. 188
4.10 Best performing combinations in terms of PPV for the FN SNP experiment. 190
4.11 Best performing combinations in terms of Sensitivity for the FN SNP experiment. 191

A.1 Number of 150 bp paired-end reads generated with Sherman tool aiming 100-fold coverage for each chromosome of *A. thaliana*. 238
A.2 QUAST results for the *A. thaliana* Velvet assembly. 241
A.3 Values set for the `--score-min <func>` depending on the relaxed mismatch setting. 249
A.4 Average percentage of mapped reads for the SNP events analysed by the pipeline in each relaxed mismatch setting. 250
A.5 *A. thaliana* annotation BLASTDB hits found by the pipeline for cases with either the reference or the alternate alleles present in the reads sets considering the most relaxed mapping scenario. 251

B.1 Final optimised *A. thaliana* Velvet assembly details retrieved from the VelvetOptimiser.pl logging information. 263
B.2 QUAST results for the *A. thaliana* optimised Velvet assembly. . . . 263

C.1 Original genome sequence statistics. 271
C.2 Coverage depth calculations. 271
C.3 Number of reads per read length dataset. 271
C.4 Insert lengths and standard deviations for each read length dataset. 272
C.5 QUAST results for each assembly replicate. 284
C.6 Values for BWA-SW -T parameter (last column) – 2% mismatches. 287
C.7 Values for BWA-SW -T parameter (last column) – 14% mismatches. 288
C.8 *Arabidopsis thaliana* annotation general composition *versus* unique SNP manifests. 301

List of Figures

1.1 Variants among homologous DNA sequences. 7
1.2 Typical SNP calling pipeline workflow. 13
1.3 454 technology. 16
1.4 Solexa (and later Illumina) technology. 17
1.5 ABI SOLiD^{TM} technology. 18
1.6 Sanger and NGS technologies examples. 23
1.7 TGS examples. 24
1.8 Repetitive sequences of the genome can complicate assemblies. . . . 32
1.9 Ploidy, heterozygosity and their impact on the assembly. 33
1.10 de Bruijn graph structure for assembly. 37
1.11 Ambiguity in read mapping. 41
1.12 Examples of algorithmic approaches to tackle the NGS short-read mapping problem. 43
1.13 PCR duplicates and an example strategy for remediation. 45
1.14 Example of a probabilistic method in single nucleotide variant calling from NGS data. 50
1.15 Example of base-calling error bias and improvement effect after countermeasure applied. 53
1.16 Types of errors. 55
1.17 Optical duplicate and a resulting FP SNP. 56
1.18 Reference misassembly and a resulting FP SNP. 57
1.19 Read mismapping and resulting FP SNPs. 59
1.20 Example of misalignment. 60

2.1 Screenshot of an alternate allele being present in significant minority at a given SNP locus. 64
2.2 Example of a homozygous SNP visualised with the Tablet tool. . . 65
2.3 Tablet screenshots of a SNP in the outputs of different mapping modes of the Bowtie tool. 66
2.4 Schematic of reads misplaced during assembly, caused by a two copy repeat R. 68
2.5 The putative mechanism for reference misassembly origination. . . . 69

2.6 The pipeline workflow for the Bowman RNA-Seq dataset experiment. 77
2.7 Screenshots taken with the Tablet graphical viewer showing the mappings obtained for the reference transcript comp13964_c0_seq2 at position 950. 79
2.8 The refactored pipeline workflow for the *A. thaliana* dataset experiment. 88
2.9 Screenshots taken with the Tablet graphical viewer showing the mappings obtained for the reference contig NODE_6286 at position 172. 90
2.10 Screenshot showing the labels of reads revealed by the relaxed mapping (with 5 mismatches allowed) for contig NODE_6286 at position 172. 91
2.11 Screenshot showing the AFG file portion related to contig NODE_6286 with three distinct classes of reads taking part in the assembly. . . . 92
2.12 Screenshots showing three different observed scenarios for contig NODE_6286, placed side by side for comparison purposes. 93
2.13 Screenshots showing three different scenarios for contig NODE_18482, placed side by side for comparison purposes. 94
2.14 Screenshots showing three different scenarios for contig NODE_3278, placed side by side for comparison purposes. 95
2.15 Screenshots showing the cases undetected or only partially detected by the pipeline. 97
2.16 Screenshots showing three different scenarios observed for contig NODE_19266, placed side by side for comparison purposes. 98
2.17 Screenshots showing three different scenarios observed for contig NODE_8802, placed side by side for comparison purposes. 99
2.18 Percentages of events with BLAST hits in the increasing databases of Haruna Nijo FLcDNAs. 105
2.19 Percentages of events with BLAST hits considering a continuous increase in the sizes of the databases of Haruna Nijo FLcDNAs. . . 106
2.20 Misassembly due to different haplotypes. 107

3.1 Comparison between the Bowtie tool "unique" and *all* mapping modes visualised with the Tablet tool. 113
3.2 Tablet screenshots illustrating the comparison between the Bowtie mapping modes *all* (left), "unique" (middle), and "unique" plus flags `--best --strata` (right). 115
3.3 Tablet screenshots illustrating the comparison between the Bowtie tool behaviours with and without the `--best --strata` flags applied. 117
3.4 Visualisation of mismapped reads with Tablet tool. 118
3.5 A FP SNP/false heterozygosity scenario exemplified with Tablet screenshots. 120

3.6 Control conceptualised. . . . 123
3.7 An example of mismapping identified by the read origin information available from the read simulator. . . . 124
3.8 Workflow of the mismapping quantification code. . . . 126
3.9 Design of the read mismapping proof-of-concept experiment. . . . 127
3.10 Read mismapping experiment annotation results. . . . 129
3.11 FP SNP sites genomic locations (*de novo* assembly). . . . 130
3.12 FP SNP sites genomic locations (control genome). . . . 130
3.13 Representation of the *A. thaliana* chromosomes. . . . 135

4.1 Experimental design. . . . 151
4.2 Read mismapping quantification code workflow. . . . 157
4.3 Tools and variables used in the experiment. . . . 159
4.4 5-way interaction between assembly, mapper, read length, MAPQ, and mapping stringency. . . . 163
4.5 5-way interaction between assembly, mapper, variant caller, MAPQ, and read length. . . . 164
4.6 SNP annotation. . . . 172
4.7 FP SNP sites genomic locations (first Velvet *de novo* assembly replicate). . . . 173
4.8 Mismatches *versus* read length. . . . 180
4.9 Tablet screenshots of read mismapping and corresponding FP SNPs. 181
4.10 Percentages of mismapped reads as a function of read length and type of reference assembly. . . . 183
4.11 BamQC tool screenshot of a 'BWA-STRICT' mapping of the FN SNP experiment. . . . 193
4.12 BamQC tool screenshot of a 'Bowtie-STRICT' mapping of the FN SNP experiment. . . . 193

C.1 General composition of the *Arabidopsis thaliana* annotation. . . . 302
C.2 BLAST-based annotation results for the SNP manifests from the Allpaths-LG first replicate of the experiment. . . . 302
C.3 BLAST-based annotation results for the SNP manifests from the Allpaths-LG second replicate of the experiment. . . . 303
C.4 BLAST-based annotation results for the SNP manifests from the Velvet first replicate of the experiment. . . . 303
C.5 BLAST-based annotation results for the SNP manifests from the Velvet second replicate of the experiment. . . . 304
C.6 BLAST-based annotation results for the SNP manifests from the two controls (compiled) of the experiment. . . . 304

C.7 FP SNP sites genomic locations (first Allpaths-LG *de novo* assembly replicate). 305
C.8 FP SNP sites genomic locations (second Allpaths-LG *de novo* assembly replicate). 306
C.9 FP SNP sites genomic locations (first Velvet *de novo* assembly replicate). 306
C.10 FP SNP sites genomic locations (second Velvet *de novo* assembly replicate). 307
C.11 FP SNP sites genomic locations from the compiled controls of the experiment. 307

Acknowledgements

Dr Micha Bayer, Prof. Andrew Flavell, and Dr David Marshall as my supervisors. Thank you for the guidance and support. Thank you very much, Micha, for your friendship, the invaluable Java training in the beginning of the project and all the suggestions throughout.

Agnieszka Golicz for the groundwork on false positive SNPs. All the colleagues of the JHI Dundee ICS group for the support and friendship. Special thanks to Iain Milne and Gordon Stephen for the continuous development of the Tablet assembly viewer and Corran Musk for the computing cluster setup. Sebastian Raubach for the help with Java, Wenbin Guo and Sebastian (again) for the LaTeX tips, and Ewan Mollison for the insights into Biology. Christine Hackett for the help with the data statistical analysis. June Johnston for all the administrative assistance during my time in JHI and the printing of this material.

Laura Logie, Craig Simpson, Peter Cock, and Colin Campbell of the JHI's Postgraduate school. The JHI, the RESAS/Scottish Government, and the University of Dundee for co-funding my studentship.

My work colleagues Brennan Martin, Eduardo Paiva, Matthew Gemmell, Sophie Shaw, and my line manager Elaina Collie-duguid, at University of Aberdeen, for their understanding and flexibility during my hectic first year period due to the thesis write-up.

Our families for the encouragement. My father Abel, my mother-in-law Marilene, my sister-in-law Carla, and my brother Marcelo for all the support while we are away from Brazil. My mother Vera (*In Memoriam*) for everything.

To my mate-pair read Gabriela and our reference sequences Vitória and Laszlo, to whom we always aim to map with the highest mapping quality score and accuracy to avoid false positive bonds. I will never be able to thank you enough for the countless wall-clock hours, days, and weekends not spent with you during my whole PhD. Let alone your courage to relocate to another country so I could continue to build my new career in Bioinformatics. You are BRAVEHEARTED! I LOVE YOU!

UNIVERSITY OF DUNDEE

COLLEGE OF LIFE SCIENCES

I certify that Antonio Claudio Bello Ribeiro has satisfied all the terms and conditions of the relevant Ordinance and Regulations to qualify in submitting this thesis in application for the degree of Doctor of Philosophy.

Dated: August 2016

Research Supervisors: _____
Dr Micha Bayer

_____
Prof. Andrew Flavell

_____
Dr David Marshall

UNIVERSITY OF DUNDEE

Date: **August 2016**

Author: **Antonio Claudio Bello Ribeiro**

Title: **A quantitative exploration of causes of false positive single nucleotide polymorphisms in Next-Generation Sequencing data**

Department: **College of Life Sciences**

Degree: **Ph.D.**

I hereby declare that the work described in this thesis is my own; that I am the author of this thesis; that it has not previously been put forward in submission for any other degree or qualification; and that I have consulted references herein.

Antonio Claudio Bello Ribeiro

Abstract

Single Nucleotide Polymorphisms (SNPs) are widely used molecular markers, and their use has increased massively since the inception of Next-Generation Sequencing (NGS) technologies, which allow detection of large numbers of SNPs at low cost. However, both NGS data and their analysis are error-prone, which can lead to the generation of false positive (FP) SNPs. The traditional approach to SNP discovery is based on mapping reads to a reference sequence. Apart from sequencing errors, which vary in pattern and rate depending on the sequencing platform, the short read lengths that prevail in NGS, together with the repetitive nature of the genomes of many organisms, can lead to errors in the genome assembly and/or read mapping stages of the mapping-based approach for SNP discovery.

The work described here has investigated and quantified some mechanisms that cause false positive SNPs. These include reference misassembly due to the presence of paralogous sequences and read cross-mapping, along with associated factors such as quality of the reference sequence, read length, choice of mapper

and variant caller, mapping stringency, and filtering of SNPs by read mapping quality and read depth. The study shows that both paralogs and the choice of tools and parameters involved in variant calling can have a dramatic effect on the number of FP SNPs produced. A brief exploration of the influence of these factors towards false negative (FN) SNPs generation is also carried out in the end of the study, paving the way to new insights. This thesis aims to provide a stepping stone towards a better understanding of the factors influencing the mapping-based SNP discovery approach.

List of Abbreviations

A — Adenine

ABI — Applied Biosystems

ABI SOLiD™ — Applied Biosystems Sequencing by Oligonucleotide Ligation and Detection

ABMMS — Advances in Biological and Medical Measurement Science

ABySs — Assembly By Short Sequences

AFG — Augmented fragment / Assembled fragment

AFLP — Amplified Fragment Length Polymorphism

Allpaths-LG — Allpaths-Large Genomes

alt. / altern. — alternate

AMOS — A Modular Open-Source consortium

ANNOVAR — Annotate Variation

ANOVA — Analysis of Variance

API — Application Programming Interface

ASCII — American Standard Code for Information Interchange

ATP — Adenosine triphosphate

avg. — average

BAM — Binary Alignment/Map

Bash — Bourne Again Shell

BAYSIC — BAYeSian Integrated Caller

bp — base pair(s)

BFAST — Blat-like Fast Accurate Search Tool

BiSCaP — Binomial SNP Caller from Pileup

BLAT — the BLAST-like alignment tool

BLAST — Basic Local Alignment Search Tool

BLASTDB — BLAST database

BWA — Burrows-Wheeler Aligner

BWA-SW — BWA-Smith-Waterman

BWT — Burrows-Wheeler transform

C — Cytosine

CABOG — Celera Assembler with Best Overlap Graph

CAP3 — Contig Assembly Program 3

cDNA — Complementary DNA

CDS — Coding sequence

ChIP — Chromatin Immunoprecipitation

Chr / chr — Chromosome

Chr1 — Chromosome 1

Chr2 — Chromosome 2

Chr3 — Chromosome 3

Chr4 — Chromosome 4

Chr5 — Chromosome 5

Chrm / chrm. — Chromosome

CNAG — Centro Nacional de Análisis Genómico

CNV — Copy-Number Variation/Variant

COST — European Cooperation in Science and Technology

cov. — covering / coverage

CPU — Central Processing Unit

d.f. — degrees of freedom

DNA — Deoxyribonucleic acid

DNA-Seq — DNA sequencing

EBI — The European Bioinformatics Institute

Edena — Exact *De Novo* Assemble

ELAND — Efficient Large-Scale Alignment of Nucleotide Databases

EMBL — European Molecular Biology Laboratory

EMBOSS — The European Molecular Biology Open Software Suite

emPCR — emulsion PCR

EST — expressed sequence tag

F — Force

FB — FreeBayes

F prob. — F probability

FLcDNA — Full-length cDNA

FN — False negative

FN SNP — False negative SNP

FP — False positive

FP SNP — False positive SNP

G — giga

G — Guanine

G2P — gen2phen

GAGE — Genome Assembly Gold-standard Evaluations

GATK — Genome Analysis Toolkit

Gbp — giga base pairs

GBS — Genotyping-by-Sequencing

GCAT — Genome Comparison and Analytic Testing

GMS — Genome Mappability Score

h — hour(s)

HapMap — haplotype map

HRM — High-Resolution Melting

HTS — High-Throughput Sequencing

IBGSC — The International Barley Genome Sequencing Consortium

IBM — International Business Machines

ICS — Information and Computational Sciences

ID — Identifier

IGV — Integrative Genomics Viewer

Inc. — Incorporated

indel — insertion/deletion

int. — intergenic (region)

IUPAC — International Union of Pure and Applied Chemistry

IWGSC — The International Wheat Genome Sequencing Consortium

Java SE — Java Standard Edition

JGI — Joint Genome Institute

JHI — The James Hutton Institute

k — kilo

KASP — competitive allele-specific PCR

kb — kilo bases

kbp — kilo base pairs

LCR — Low-complexity region(s)

LD — Linkage Disequilibrium

LTD. / Ltd. — Limited liability company

LoQuM — LOgistic regression tool for calibrating the Quality of short read Mappings

M — mega

MAPQ — Mapping quality

MAQ — Mapping and Assembly with Qualities

MAS — Marker-Assisted Selection

MaSuRCA — Maryland Super-Read Celera Assembler

max. — maximum

Max. — Maximum

Mb — mega base(s) / mega base pairs

Mbp — mega base pairs

Mbp — million base pairs

MHC — Major histocompatibility complex

MIRA — Mimicking Intelligent Read Assembly

mRNA — Messenger RNA

MRP — Main Research Provider

m.s. — mean square

MT/CP — mitochondrial and chloroplast insertions

n — number of mismatches allowed

NCBI — National Center for Biotechnology Information

NGS — Next-Generation Sequencing

NIH — National Institutes of Health

NIST — National Institute for Standards and Technology

NP-hard — Non-deterministic Polynomial-time hard

OLC — Overlap-Layout-Consensus

p. — page

PacBio / PACBIO — Pacific Biosciences

PCR — Polymerase Chain Reaction

PE — Paired-end / Paired-ended

Percentage SS — Percentage of sum of squares

PerM — Periodic Seed Mapping

perm. prob. — permutation probability

Phred — PHil's Read EDitor

pos. — position

Pp. — pages

PPV — Positive Predictive Value

QA — Quality Assessment

QSRA — Quality-value guided Short Read Assembler

QUAST — QUality ASsessment Tool

RAPD — Randomly Amplified Polymorphic DNA

rDNA — Ribosomal DNA

ref. — reference

RESAS — Rural & Environment Science & Analytical Services

RFLP — Restriction Fragment Length Polymorphism

RNA — Ribonucleic acid

RNA-Seq — RNA sequencing

SAM — Sequence Alignment/Map

SBH — Sequencing-by-Hybridization

SBS — Sequencing-by-Synthesis

SE — Standard Error

Sed — Standard error of the difference

Seq-based — Sequencing-based

SGA — String graph assembler

SGS — Second-Generation Sequencing

SHARCGS — SHort Read Assembler based on Robust Contig extension for Genome Sequencing

SHRiMP — SHort Read Mapping Package

SInC — Snp, Indel and Cnv

SMRT — Single-Molecule, Real-Time

SNP — Single Nucleotide Polymorphism

SnpEff — SNP Effect

SNV — Single Nucleotide Variation/Variant

SOAP — Short Oligonucleotide Analysis Package

SOAP2 — Short Oligonucleotide Analysis Package 2

SOAPdenovo — Short Oligonucleotide Analysis Package *de novo*

SOAPsnp — Short Oligonucleotide Analysis Package SNP

SOCS — Short Oligonucleotide Color Space

SPBAU — St. Petersburg Academic University

s.s. — sum of squares

SSAHA — Sequence Search and Alignment by Hashing Algorithm

SSAHA2 — Sequence Search and Alignment by Hashing Algorithm 2

SSAKE — Short Sequence Assembly by K-mer search and 3' read Extension

ssDNA — Single-stranded DNA

SSE — Sequence-Specific Error

SSR — Simple Sequence Repeats

st. dev. — standard deviation

T — Thymine

TAIR — The Arabidopsis Information Resource

TE — transposable element

t.e.g. — transposable element gene

TGAC — The Genome Analysis Centre

TGS — Third-generation Sequencing

TP — True positive

UC Davis — University of California, Davis

UGT — Unified Genotyper

USA — United States of America

USDEOS — United States Department of Energy – Office of Science

VCAKE — Verified Consensus Assembly by K-mer Extension

VCF — Variant Call Format

v.r. — variance ratio

vs — *versus*

ZMW — Zero-Mode Waveguide

ZOOM — Zillions Of Oligos Mapped

$\sim$ — approximately

$\neq$ — different

\# — Number / Number of

% — Percentage

Chapter 1

Literature Review

Preface

The aims of this thesis are to investigate the existence and quantify the impact of some of the mechanisms behind false positive (FP) Single Nucleotide Polymorphisms (SNPs) (loci incorrectly identified as polymorphic), specifically those observed in mapping-based SNP calling approaches. Factors that can potentially influence the magnitude of these FP events, e.g. read length, are also considered. Finally, some guidance for avoiding such FP SNP occurrences is provided, but this has not been the final objective of the work and neither was the benchmarking of tools associated to the process.

Chapter 1 aims to provide a broad technical background for the thesis by introducing the main aspects of the traditional variant discovery procedure utilising Next-Generation Sequencing (NGS) data. The main focus is on SNP discovery, particularly in non-model organisms, and some of the artefacts that may arise

during this process (e.g. FP SNPs).

Chapter 2 describes the experiments which were carried out to determine the extent to which misassembly of the reference sequence leads to homozygous FP SNPs. This was based on the assumption that the presence of such SNPs, in a mapping of reads against a reference that was *de novo* assembled from the same reads, is a possible indication of misassembly of the reference sequence at the location of the homozygous SNP. An *in silico* pipeline was then developed to analyse such homozygous SNP events in RNA-Seq data from the barley cultivar Bowman and in simulated genomic reads from the ∼125 Mbp genome of the flowering plant *Arabidopsis thaliana*. The approach aimed to test whether such SNPs can be caused by misassembly of the reference sequence due to the existence of multiple, related but distinct sequences (e.g. paralogs) used in the assembly.

Chapter 3 redirects the focus to heterozygous FP SNPs. Taking advantage of the NGS data processing workflow simulation based on the *A. thaliana* genome, it investigates the mechanism of read mismapping and the consequent occurrence of such kind of FP SNPs along with providing corresponding quantification.

Based on the read mismapping concept described in Chapter 3, Chapter 4 goes on to investigate the impact of other factors and their interactions on the FP SNP generation. Thus, the exploration of other tools, parameters, and other potential causative factors of FP SNPs is carried out still taking advantage of the NGS

workflow simulation. More specifically, NGS read datasets, varying in length from 50 to 1,000 bp, are used to generate both new genome assemblies and mappings to test the effects of NGS read length, different software for genome assembly, read mapping, and SNP calling (including variable parameter settings) stages, as well as SNP filtering, on FP SNP generation.

Finally, Chapter 5 provides general conclusions and briefly outlines opportunities for future work motivated by this study.

1.1 Introduction

The study of organisms' genomes is the subject of the field known as *Genomics* (Hartl, 2011; Zhang et al., 2011). Aiming to understand the molecular organisation of genes in an organism as well as how they function, interact, and evolve (Hartl, 2011), genomics has been impacting science and society, unveiling knowledge not only related to the human genome but also to those of other organisms, confirming its anticipated applicability (USDEOS, 2008; Kahvejian et al., 2008). As an example, it has been applied to better use crop genetic resources, aiming to expand the overall knowledge about plant biology, especially in terms of breeding and improvement (Dhanapal, 2012). Better yields and resistance to challenging environmental conditions as well as to pathogens are examples of the motivation behind this.

Since its commercial launch in 2005 (Kling, 2005; Coombs, 2008; Margulies

et al., 2005) and throughout the past decade, Next-Generation Sequencing (NGS), also referred as second-generation sequencing (SGS) (Henson et al., 2012) or sometimes by the more general term high-throughput sequencing (HTS) (Altmann et al., 2012), has accelerated the acquisition of biological data in an unprecedented manner. Shifting the bottleneck from data generation to its analysis (Kahvejian et al., 2008; Schuster, 2008; Mardis, 2010), NGS keeps promoting the translation of such data into genomics knowledge and, consequently, generating tangible benefits for fundamental and applied research, including personalised medicine (Kling, 2005; Auffray et al., 2009). The number of 'Seq-based' applications already in place is vast (Morozova and Marra, 2008; Wold and Myers, 2008): *de novo* genome assembly (Henson et al., 2012; Yandell and Ence, 2012), *de novo* transcriptome assembly (Robertson et al., 2010; Martin and Wang, 2011; Grabherr et al., 2011), metagenomics (Qin et al., 2010; Ruffalo et al., 2011), pharmacogenomics (Henson et al., 2012), phylogenomics (Lu et al., 2014), cancer genomics (Guffanti et al., 2009; Ruffalo et al., 2011), analysis of mRNA expression data (Sultan et al., 2008; Ruffalo et al., 2011; Anders et al., 2013), DNA methylation studies (Taylor et al., 2007; Laird, 2010; Ruffalo et al., 2011; Krueger et al., 2012), Chromatin Immunoprecipitation (ChIP) studies (Barski et al., 2007; Johnson et al., 2007; Wold and Myers, 2008), detection of genomic structural variants (Alkan et al., 2009; Medvedev et al., 2009; Ruffalo et al., 2011; Lam et al., 2012), detection

of variants/SNPs and Genotyping-by-Sequencing (GBS) (Mardis, 2008; Li et al., 2009b; Liao and Lee, 2010; Nielsen et al., 2011; Altmann et al., 2012; Poland and Rife, 2012), among others.

One highly relevant application among those mentioned is the discovery of DNA sequence variants. Considering a given genomic reference sequence for comparison, variants can be defined as variations in an individual's DNA. Broadly speaking, they can be divided into *sequence variants* — SNPs, short nucleotide insertions and/or deletions and substitutions — and *structural variants* — insertions, deletions, duplications, copy numbers, inversions, and translocations (Scherer et al., 2007; Medvedev et al., 2009; Ruffalo et al., 2012; Ensembl, 2016). The latter are responsible for large-scale changes in an organism's chromosome structure and typically encompass events >1,000 base pairs (bp) in length (Scherer et al., 2007). However, they can also be categorised by how they affect the copy count of any genomic region (e.g. insertions and deletions are referred to as copy-number variants (CNVs); inversions are copy-count invariant) (Medvedev et al., 2009). On the other hand, sequence variants usually characterise changes of few nucleotides. In these terms, for instance, a SNP can be defined as a single-base difference between a reference sequence and a sample, or among a number of samples (e.g. a thymine base *versus* a guanine base). Small indels, which can also be classified as a particular category of SNPs (Duran et al., 2009), represent the insertion or

deletion of a small number of bases in samples relatively to a reference sequence (Kunda, 2015). A reference sequence is intended to provide a "gold standard" that represents a given organism's genome/transcriptome/exome, and may either be a publicly available resource or, alternatively, may have been generated for the purpose of a specific project only (Kunda, 2015; Yi et al., 2010; Altmann et al., 2012).

Slight variations in an individual's DNA sequence are important because they can have a major impact on disease development, behaviour or response to environmental factors, like infectious microbes, toxins, conditions, and drugs (USDEOS, 2008), when they occur within a gene or in a regulatory region near a gene. This is true irrespective of the domain of life the species of interest belongs to.

However, variations in an organism's genome may not necessarily have an effect on its phenotype — "the physical manifestation of genetic information" (Liao and Lee, 2010). In fact, the majority of this type of variation is observed in regions that do not code for proteins. Even variations in non-coding regions can be used as a genetic (or biological/molecular) marker and hence those genetically linked to genes implicated in phenotypes of interest are particularly important (Hartl, 2011). "These markers are informative signposts distributed throughout the genome at the highest resolution feasible at the time" (Kahvejian et al., 2008).

According to Hartl (2011), a genetic difference (or variation) that is relatively common in a population — a group of individuals of the same species (Griffiths et al., 2012) — is called a *polymorphism* (Figure 1.1).

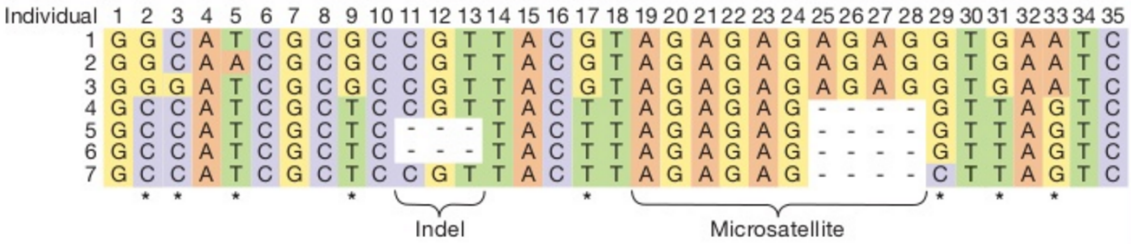

Figure 1.1: Variants among homologous DNA sequences, as described by Griffiths et al. (2012). Variation present in the aligned DNA sequences of seven individual chromosomes is illustrated. The '*' (asterisk) symbols show the locations of SNPs. A 3 bp indel and a 4 bp microsatellite polymorphism are also shown. Numbers on the foremost left column indicate the chromosomes. Numbers on the top row indicate the base pair position. Adapted from Figure 18-1 (Griffiths et al., 2012). Permission from: An Introduction to Genetic Analysis 11E, by Anthony J.F. Griffiths, et al, Copyright 2015 by W.H. Freeman and Company. Used by Permission of the publisher.

The most common type of sequence variation in most genomes is the SNP (USDEOS, 2008; Liao and Lee, 2010; Griffiths et al., 2012). By this definition, SNPs are sites where typically more than 1 percent of individuals in a population differ in their DNA sequence (USDEOS, 2008; Liao and Lee, 2010; Hartl, 2011). More loosely, such sequence alternatives are referred as "alleles" (Brookes, 1999; SNPedia, 2008) — which in fact are formally defined as "alternate forms of a gene of chromosomal locus that differ in DNA sequence" (Liao and Lee, 2010). Most SNPs are bi-allelic, meaning that only two of the four common nucleotides are encountered in a population of individuals in a specific position (Brookes,

1999; Vignal et al., 2002; Liao and Lee, 2010; Griffiths et al., 2012). They can be classified as *transitions* (A-G or C-T) — when the base is replaced by another of the same category (e.g. a purine by a purine or a pyrimidine by a pyrimidine) — or *transversions* (A-C, A-T, C-G or G-T) — when the base is replaced by another of the opposite category (e.g. a purine by a pyrimidine or vice-versa — (Lai et al., 2012). When a SNP falls in a protein-coding region, it can be of two types: *synonymous* or *non-synonymous*. The first type does not affect the encoded protein because the base change produces a codon which encodes the same amino acid as the original one. The non-synonymous type produces a different codon and hence a different coded protein, being classified into two other subtypes: *missense* and *nonsense*. In the former case, there is the substitution of an amino acid for another. In the latter, the original codon is changed into a stop codon and almost always this results in loss of gene function (Hartl, 2011).

Griffiths et al. (2012) state that SNPs are usually considered *common* in a population if the less common allele occurs at a frequency of about 5 percent or greater while being considered *rare* if the less common allele occurs at a frequency lower than that. In humans, for example, there were identified millions of locations where single-base DNA differences occur (roughly 10 million or a common SNP about every 300 to 1,000 bp) (NIH, 2007; HapMap, 2003). There are even greater numbers of rare SNPs (Griffiths et al., 2012). As an addendum, very often,

polymorphisms are not independent of one another, meaning that, when a mutation arises, it is associated with particular variants present on the same chromosome (Goldstein and Cavalleri, 2005). Such variants that associate with each other are known as a 'haplotype' (Goldstein and Cavalleri, 2005) — "a set of alleles located at neighbouring genes or genomic sequences that tend to be inherited together" (Liao and Lee, 2010). As explained by Griffiths et al. (2012), for example, two homologous chromosomes sharing the same allele at each of the loci in question have the same haplotype. If two chromosomes present distinct genotypes at even one of the loci, then they have different haplotypes.

Due to their large numbers in virtually all species, SNPs are currently the marker of choice, being applied in diverse areas of research which range from human forensics to resource management in fisheries (Dou et al., 2012; Kumar et al., 2012). SNPs have also been extensively used as biomarkers in human disease genetics, plant and animal breeding, population genetics, and pharmacogenetics (Morin et al., 2004; Liao and Lee, 2010). Apart from being the most abundant type of molecular genetic marker (Lai et al., 2012), SNPs may be considered the ultimate genetic marker, as they represent the highest resolution of a DNA sequence and have a low mutation rate (Lorenc et al., 2012). In plants, markers such as isoenzymes, restriction fragment length polymorphisms (RFLPs), randomly amplified polymorphic DNA (RAPD), amplified fragment length polymorphisms

(AFLPs), and single sequence repeats (SSRs; also known as microsatellites), have been used in breeding and related research since the 1920s (Duran et al., 2009; Henry and Edwards, 2009; Edwards and Henry, 2011). More recently, SNPs have become key players in crop improvement and breeding programs, as molecular markers associated with specific agronomic traits of interest (e.g. yield, disease resistance, drought resistance, product quality, etc.). Such molecular genetic markers are based on variations in the genome which can be scored between individuals and across generations (Edwards and Batley, 2010). This also allows the assessment of genetic diversity within and between related species (Lai et al., 2012). The identification of a gene underlying a trait enables the transfer of that gene between cultivars and even species using genetic modification or, alternatively, desirable alleles conferring the trait may be incorporated into a cultivar by marker-assisted selection (MAS) breeding (Edwards and Batley, 2010).

However, in order to study and explore SNPs, it is first necessary to determine which sites in the genome are variable, in a step called *SNP discovery*. Once SNPs have been discovered, the genotype — the inheritable genetic constitution of an organism (Liao and Lee, 2010) — in terms of allelic composition of different individuals in the population at each SNP site can be determined (Griffiths et al., 2012). This process is called *genotyping*.

As an example, the ever increasing throughput of NGS enables *de novo* and

reference-based large scale SNP discovery with reduced associated cost for many plant species (Kumar et al., 2012). Traditional laboratory-based methods have given way to *in silico* approaches largely because of this dramatic increase in sequence throughput. Having aligned the read fragments of one or more individuals to a reference genome, 'SNP calling' identifies variable sites, whereas 'genotype calling' determines the genotype for each individual at each site (Nielsen et al., 2011). The major challenge in variant discovery, though, is the ability to distinguish real polymorphisms from artefacts arising either in the sequencing process or the downstream bioinformatic analysis (Duran et al., 2009). NGS data is generally of lower quality when compared with the traditional Sanger approach. This is, however, at least partially offset by the increased throughput (Lai et al., 2012; Lorenc et al., 2012).

The analysis of the resulting wealth of sequence data has proven to be difficult and provides a range of significant challenges, where the central and most basic one is the correct deduction of the sequence under investigation from the very short sequence fragments containing a significant number of errors. The repetitive nature of the genomes of many organisms (Dou et al., 2012) promotes consequent mis-handling of these repetitive sequences during the genome assembly and/or mapping of reads to assemblies, leading to downstream analysis errors. These and other issues can result in a significant number of false positive, as well as false

negative, SNPs.

The traditional NGS-based approach to SNP discovery relies on mapping reads to a reference sequence. More details about such process as well as causes of false positive (FP) SNPs are provided ahead in this chapter.

1.2 Main aspects of the traditional NGS-based approach to SNP discovery

The extraction of SNPs from the raw high-throughput DNA sequencing involves many processing steps and the application of a varied set of bioinformatic tools (Pabinger et al., 2014). Those comprise the following: data quality control, assembly (Kumar et al., 2012; Leggett and MacLean, 2014) and/or mapping of the NGS reads to a reference genome (sequence), post-processing of the mapping and visualisation, the SNP calling procedure itself along with SNP candidates filtering, validation, and annotation. Each of these steps contributes to the accuracy of the final SNP and genotype calls (Nielsen et al., 2011; Altmann et al., 2012). In the following subsections, a general outline is provided of the typical mapping-based pipeline and the main elements involved within it (Figure 1.2).

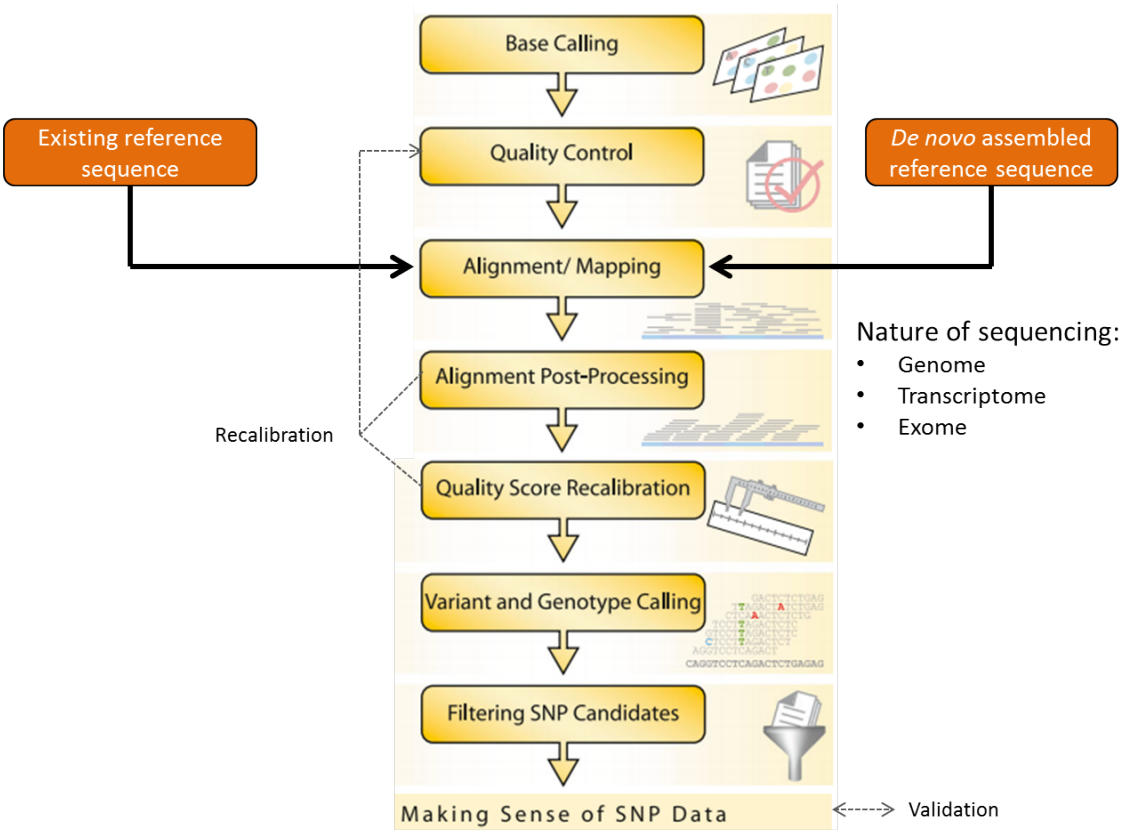

Figure 1.2: Typical SNP calling pipeline workflow. Adapted from Figure 1 "Workflow of the SNP calling pipeline" (Altmann et al., 2012). Permission from: Springer Human Genetics, A beginners guide to SNP calling from high-throughput DNA-sequencing data, 131, 2012, p. 1545, A. Altmann, et al, Copyright 2012 by Springer-Verlag. With permission of Springer.

1.2.1 Sequencing

DNA sequencing was established independently by Maxam and Gilbert (1977) and Sanger et al. (1977). Sanger sequencing became the dominant mode, adopting fluorescence-based electrophoresis methodology (Li et al., 2009b; Pop, 2009); leading to a number of monumental accomplishments, like the human genome sequence draft (Metzker, 2010). Significant improvements throughout this period — e.g.

introduction of fluorescent dye terminators of different colors (Smith et al., 1986; Prober et al., 1987; Deschamps and Campbell, 2009), replacement of the original slab gel electrophoresis by capillary separation (Luckey et al., 1990; Swerdlow and Gesteland, 1990; Deschamps and Campbell, 2009), reductions in reactions volumes — allowed the Sanger method to achieve higher throughputs, reduced reagent costs, and read lengths of up to 1,000 bp, at a per base error rate as low as 0.001% (Smith et al., 1986; Ewing and Green, 1998; Edwards and Henry, 2011). In fact, the process has undergone "a steady metamorphosis from a cottage industry into a large-scale production enterprise requiring a specialised and devoted infrastructure of robotics, bioinformatics, computer databases, and instrumentation" (Mardis, 2008), sometimes in factory-like sequencing centers housing hundreds of sequencing instruments operated by cohorts of personnel (Schuster, 2008).

Despite continued improvements over time, the limitations of automated Sanger sequencing (Ansorge, 2009) still showed a need for new and improved technologies for large scale sequencing projects (Schuster, 2008; Metzker, 2010). For instance, although being capable of sequencing up to hundreds of samples in parallel, automated capillary sequencing remains relatively labour intensive, costly, and time consuming (Lu et al., 2014). Thus, the need for rapidness and lower costs prompted the development of new technologies known as Next-Generation Sequencing (NGS) (Droege and Hill, 2008; Mardis, 2008; Shendure and Ji, 2008; Ansorge,

2009; Metzker, 2010; Kumar et al., 2012; Liu et al., 2012; Mardis, 2013). Initially, different techniques and biochemistry principles were developed and competed with each other (Chaisson et al., 2004; Hutchison, 2007; Shendure and Ji, 2008): 'pyrosequencing' (Nyrén and Lundin, 1985; Nyrén et al., 1993; Ronaghi et al., 1996; Hyman, 1988; Ronaghi et al., 1998; Dressman et al., 2003), multiplex polony sequencing (Mitra et al., 2003; Shendure et al., 2005), sequencing-by-hybridization (SBH) (Bains and Smith, 1988; Drmanac et al., 1989; Preparata and Upfal, 2000), 'sequencing-by-synthesis' (SBS) with addition and detection of the incorporated base using reversible terminators (Ansorge, 1991, 2009), sequencing from compomers (Böcker, 2004), and single-molecule sequencing (Braslavsky et al., 2003). But, from all of these endeavours, only three pioneering massively parallel sequencing techniques became commercial products and began to share the market as successful NGS solutions: 454 Life Sciences/Roche Diagnostics, with its *pyrosequencing* approach ((Margulies et al., 2005; Droege and Hill, 2008; Rothberg and Leamon, 2008); Figure 1.3), Solexa/Illumina (Bennett, 2004; Bennett et al., 2005; Bentley et al., 2008), based on the *sequencing-by-synthesis* chemistry and employing proprietary labelled reversible terminator nucleotides (Figure 1.4), and ABI SOLiD™ (Shendure et al., 2005; Valouev et al., 2008), based on the principle of *sequencing-by-ligation*, in which the DNA polymerase enzyme is swapped by a

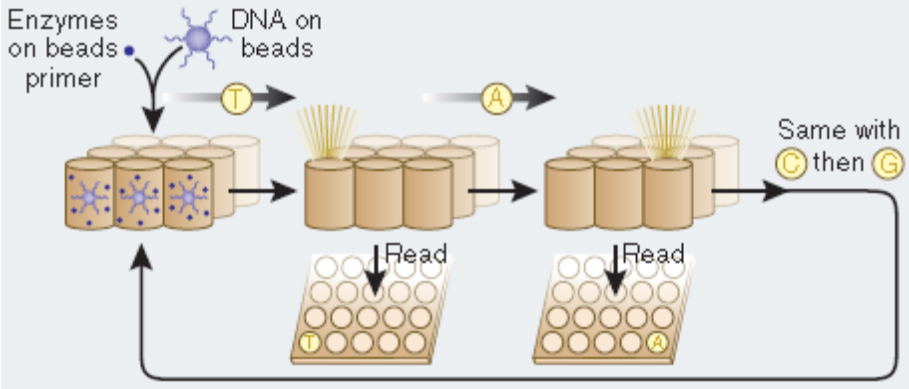

Figure 1.3: 454 technology, as detailed by Rusk and Kiermer (2008) and yourgenome.org (2015): in the sample preparation step, DNA fragments are ligated to adapters so they can be captured on beads (one fragment per bead). A water-in-oil emulsion containing PCR reagents and ideally one fragment-bead product is created so each fragment can be amplified individually per droplet. After amplification, the emulsion is broken, DNA is denatured and the beads, containing one amplified DNA fragment each, are distributed into the wells of a fiber-optic slide. The wells are loaded with enzymes and primer (complementary to the adapter on the fragment ends) needed for the sequencing reaction. Nucleotide bases are loaded to the wells in waves of one type at a time, allowing synthesis of the complementary strand of DNA to proceed. When a nucleotide is incorporated, 'pyrosequencing' takes place: pyrophosphate is released and converted to ATP, which fuels the luciferase-driven conversion of luciferin to oxyluciferin and light. As a result, the well lights up and this is recorded by a camera. The intensity of the luminosity corresponds to the number of nucleotides of the same type that have been incorporated and such pattern decoding reveals the original sequence. Adapted by permission from Macmillan Publishers Ltd: Nature Methods, Rusk and Kiermer (2008), copyright 2008.

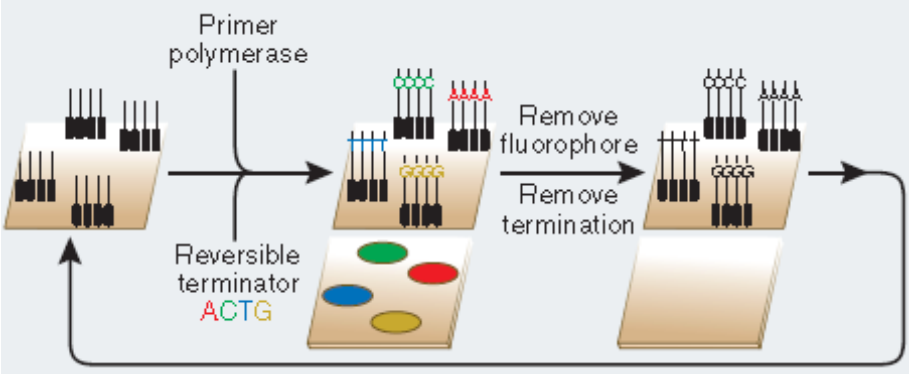

Figure 1.4: Solexa (and later Illumina) technology, as detailed by Rusk and Kiermer (2008) and Illumina, Inc. (2014b): in the sample preparation step, DNA fragments are ligated to adapters, denatured and bound at one end to a solid surface already coated with complementary adapters. Each single-stranded fragment is immobilized at one end, while its free end 'bends over' and hybridizes to a complementary adapter on the surface. The strands are clonally amplified in the presence of reagents, via the synthesis of the complementary strand, forming a double-stranded 'bridge'. Multiple cycles of this bridge amplification and subsequent denaturation create millions of clusters of single-stranded DNA copies distributed on the surface. During the sequencing-by-synthesis with reversible terminators, synthesis reagents, consisting of primers, DNA polymerase, and four differently labelled reversible terminator nucleotides, are added to the flow cell. After incorporation of a given nucleotide, a light source excites the clusters and the nucleotide is identified by its color. Then, the 3' terminator on the base and the fluorophore are removed and the cycle is repeated for a number of times that determines the read length of the sequence. Adapted by permission from Macmillan Publishers Ltd: Nature Methods, Rusk and Kiermer (2008), copyright 2008.

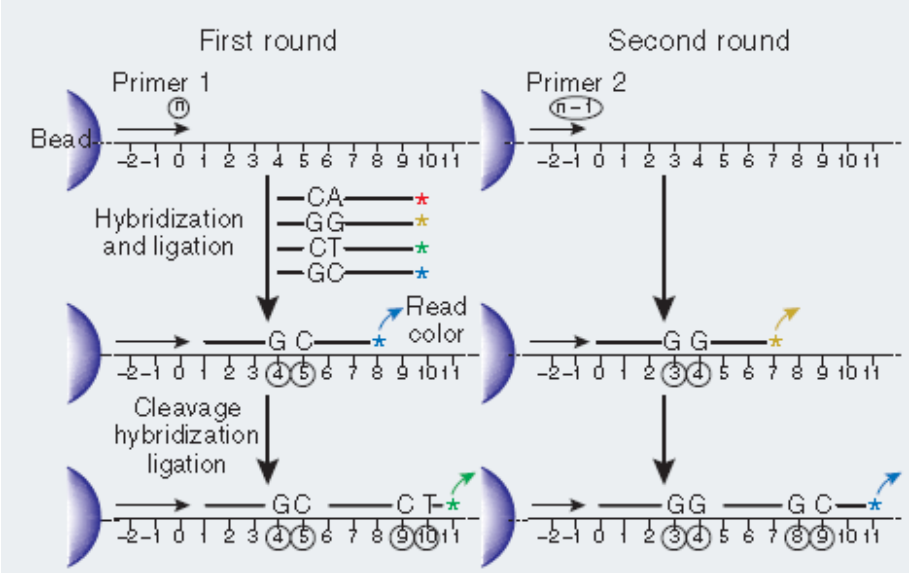

Figure 1.5: ABI SOLiD^{TM} technology, as detailed by Rusk and Kiermer (2008) and Breu (2010): in the sample preparation, DNA fragments are ligated to adapters and amplified onto beads by emulsion PCR. The DNA is denaturated and the beads are deposited onto a glass slide. The template beads are combined within a mixture consisting of an *universal* sequencing primer, ligase enzyme, and a large pool of octamer oligonucleotides (di-base probes). These di-base probes are fluorescently labelled with four dyes; each representing 4 of 16 possible di-nucleotide sequences. During the sequencing-by-ligation, the universal primer is hybridized to the adapter and its 5' end is available for ligation to an oligonucleotide hybridizing to the template sequence. The oligonucleotides compete for ligation to the primer and, when one hybridizes, it is ligated and the respective fluorescence is measured. The dye is then cleaved off and the cycle of ligation-cleavage is repeated (initially, 7 cycles yielding 35 base pairs). In the first round, the process determines possible identities of bases in positions 4, 5, 9, 10, 14, 15, etc. The synthesized strand is then removed, a new primer is hybridized (offset by one base) and the ligation cycles are repeated. The entire process is repeated to determine positions 3, 4, 8, 9, 13, 14, etc., until the first base in the sequencing primer (position 0) is reached. Since the identity of this base is known, the color is used to decode its neighbouring base at position 1, which in turn decodes the base at position 2, etc. (2-base encoding process), until all sequence pairs are identified. Adapted by permission from Macmillan Publishers Ltd: Nature Methods, Rusk and Kiermer (2008), copyright 2008.

DNA ligase one (Figure 1.5).

Despite the differences in their underlying chemistries, sequencing protocols and throughputs, all current NGS workflows share some common attributes involving: sample preparation, DNA capture (to some sort of solid surface i.e. a glass plate or microbead) followed by clonal amplification of individual molecules within the library and, finally, parallelised sequencing of the amplified library to yield, in the case of ultra-deep NGS, up to billions of 'short' sequencing reads (Pop, 2009; Lu et al., 2014). This amplification is necessary so the massively-parallel sequencing reactions are capable of emitting a sufficiently strong signal for the adequate detection through the instrument's imaging acquisition system. Sequencing is achieved via the real time microscopic image capture of the light emissions which occur during synthesis of the complementary DNA strand (Nielsen et al., 2011; Thompson and Milos, 2011; Rodríguez-Ezpeleta et al., 2012).

Another common characteristic is that the sequencing reactions are conducted in a series of repeated cycles, in a 'wash-and-scan' manner, nucleotide-by-nucleotide (Mardis, 2011; Thudi et al., 2012). Differently from the Sanger's 'sequencing-after-synthesis' method — which is based on the physical separation and detection of differently sized DNA molecules generated by the chain-termination inhibitor method in polyacrylamide gels or by capillary electrophoresis (Sanger et al., 1977) — NGS uses a 'sequencing-by-synthesis' approach. This entails real-time monitoring

of newly synthetized DNA molecules (Weber, 2015). NGS technologies' greater efficiency is also due to *in vitro* cloning and the use of solid surface systems for sequencing, instead of the laborious bacterial cloning step followed by DNA isolation in conventional Sanger sequencing (Mardis, 2011).

Depending on the platforms used, read lengths of 36 to ∼700 bp are obtained with NGS, shorter than those provided by Sanger (Pop, 2009; Lu et al., 2014). However, improvements in read length, accuracy, and throughput have been a constant feature of NGS.

Advancements for the NGS methods included not only these systems mentioned — 454 (Roche Diagnostics Corporation, 1996), Illumina (Illumina, Inc., 2009), ABI SOLiD™ (ABI, 2010) — but also the introduction of single-molecule detection approaches, also referred as third-generation sequencing (TGS) (Kumar et al., 2012) or 'next-next generation sequencing' (Schuster, 2008), which are capable of recognizing incorporation or hybridization events on single molecules (Pettersson et al., 2009; Schadt et al., 2010). SGS platforms suffer from amplification biases introduced by the PCR process and dephasing due to varying extension of templates; limitations not shared by TGS systems which eliminate the need for prior amplification of DNA (Kumar et al., 2012). TGS technologies have opened the door to read lengths of >10 kbp (Lu et al., 2014), which are expected to reduce, for instance, the complexity associated with genome assembly. However, third-generation technologies

still suffer from a higher base error rate when compared to their second-generation counterparts (Clark et al., 2013).

TGS platforms can broadly be classified into three different categories: (i) SBS, where individual nucleotides are observed as they incorporate (Pacific Biosciences single-molecule real time (SMRT) technology (Eid et al., 2009; Pacific Biosciences of California, Inc., 2015) and Life Technologies/Starlight (Beechem et al., 2015) and Ion Torrent^{TM} (Rothberg et al., 2011; Thermo Fisher Scientific, Inc., 2015)), (ii) nanopore sequencing, where single nucleotides are detected as they pass through a nanopore (Oxford Nanopore; (Clarke et al., 2009; Oxford Nanopore Technologies, 2008)), and (iii) direct imaging of individual molecules (IBM; Polonsky et al. (2007)) (Kumar et al., 2012).

Figure 1.6 compares traditional Sanger sequencing with the innovative stepwise approach of NGS technologies (Mardis, 2011). Figure 1.7 shows examples of TGS technologies. Table 1.1 (Lu et al., 2014) compares some features of Sanger sequencing against the most common high-throughput sequencing platforms. Additional references for regular comparison of the 'constantly evolving' HTS systems are Glenn (2011) and his updates (The Molecular Ecologist, 2014). It is important to highlight though that, as explained in these references, "error rates among platforms are not exactly compared". For instance, final error rate for Pacific Biosciences applies only to consensus sequencing for three independent

reads of the same template whereas, for Illumina, it is achieved for approximately 75 to 85% of bases (Glenn, 2011; The Molecular Ecologist, 2014).

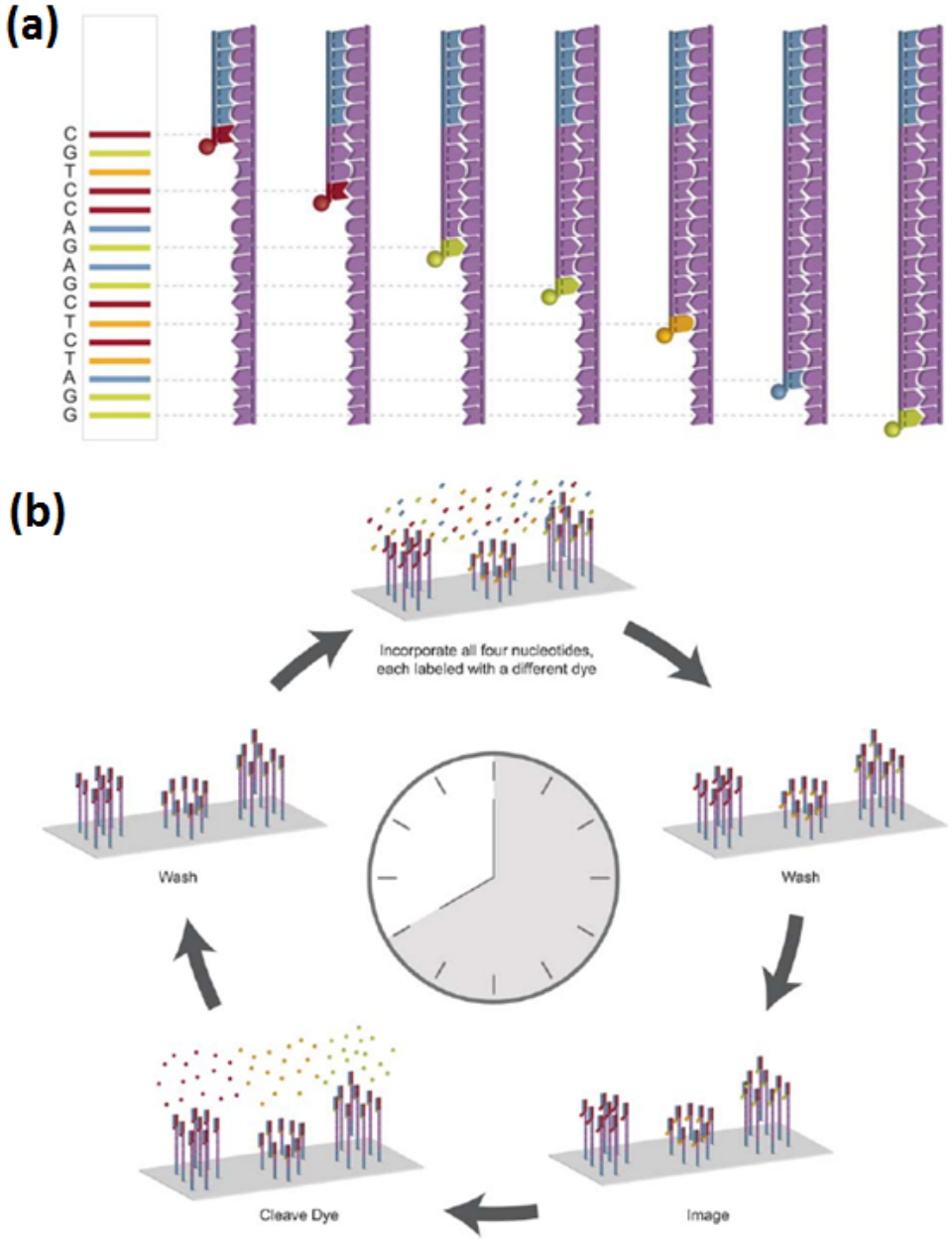

Figure 1.6: Sanger and NGS technologies examples. (a) a modern implementation of the Sanger sequencing: use of chain termination chemistry followed by size separation to resolve the sequence. (b) The Illumina process illustrates the 'step-by-step' wash-and-scan approach commonly used by NGS technologies. Adapted from Figure 1, Schadt et al. (2010), "A window into third-generation sequencing", Human Molecular Genetics, 2010, Volume 19, Review Issue 2, R227-R240, by permission of Oxford University Press.

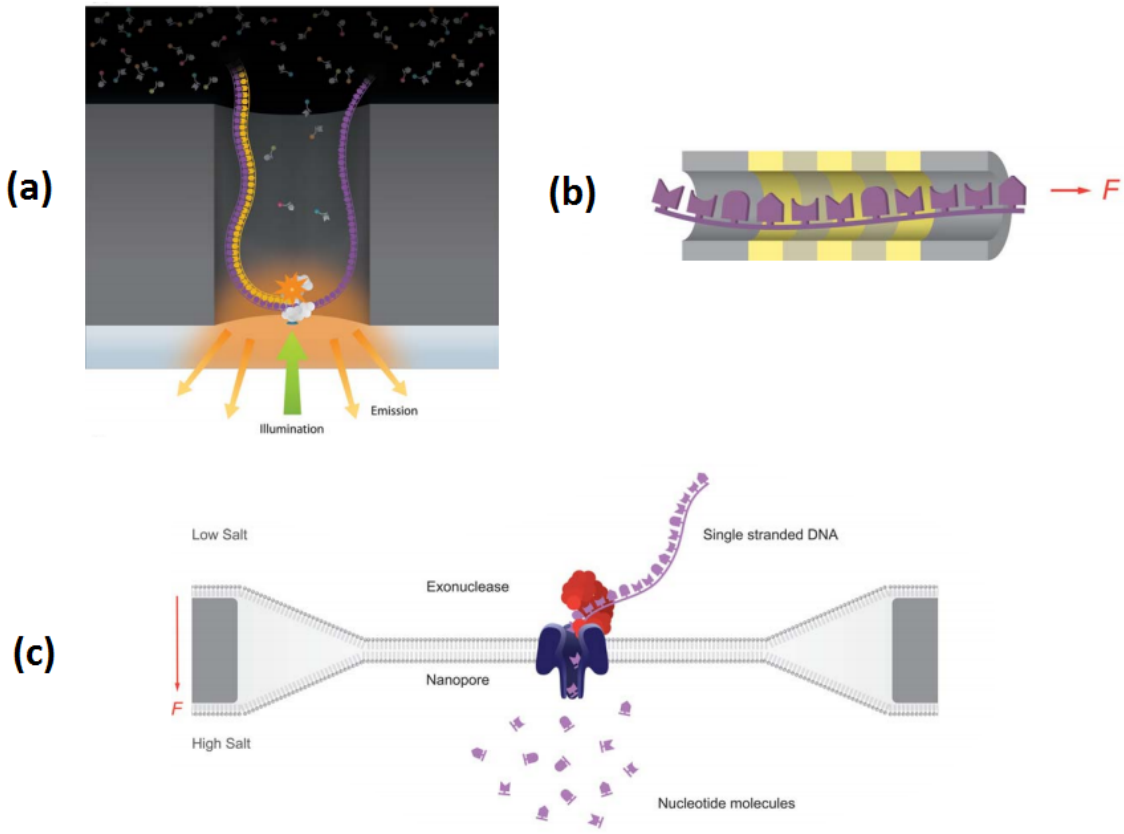

Figure 1.7: TGS examples. (a) The Pacific Biosciences technology directly observes DNA synthesis on single DNA molecules in real time. A DNA polymerase is confined in a zero-mode waveguide (ZMW). This limits illumination to a narrow region close to the polymerase. Base additions are measured with fluorescence detection of gamma-labeled phosphonucleotides. (b) Based on the unique electronic signature of each individual nucleotide, IBM's DNA transistor technology reads individual bases of single-stranded DNA (ssDNA) molecules as they pass through a narrow aperture. (c) The Oxford Nanopore technology measures the translocation of nucleotides cleaved from a DNA molecule across a pore, driven by the force of differential ion concentrations across the membrane. Adapted from Figure 2, Schadt et al. (2010), "A window into third-generation sequencing", Human Molecular Genetics, 2010, Volume 19, Review Issue 2, R227-R240, by permission of Oxford University Press.

Table 1.1: Comparison of Sanger sequencing and NGS technologies. Adapted from Table 1 (Lu et al., 2014).

Platform (vendor)	Technology	Run time	Read length per run (bp)	Max. yield	Final per base error rate (%)
Sanger (Applied Biosystems)	Chain termination	0.5-3 h	∼700-900	∼86 kbp	0.001-1.0
HiSeq/MiSeq (Illumina)	Solid-phase PCR/ reversible chain termination	4 h-11 days	36-300	15-1,800 Gbp	∼0.1
454 (Roche)	emPCR/pyrosequencing	10-20 h	∼400-700	700 Mbp	∼1
SOLiD^{TM} (Life Technologies)	emPCR/sequencing by ligation and 2-base coding	8 days	85-110	155 Gbp	≤1
Ion Torrent^{TM} (Life Technologies)	emPCR/semiconductor sequencing	2.5-7.5 h	175-400	12 Gbp	∼2
PacBio (Pacific Biosciences)	SMRT sequencing	∼0.5-3 h	∼8500	∼375 Mbp/cell	≤1

Abbreviations: bp: base pair(s); Max.: Maximum; h: hour(s); ∼: approximately; kbp: kilobase pairs; PCR: Polymerase Chain Reaction; Gbp: Gigabase pairs; emPCR: emulsion PCR; Mbp: Megabase pairs; SMRT: Single-Molecule, Real-Time.

1.2.2 Variant calling pipeline – Base calling and quality control stage

As summarised in the works of Nielsen et al. (2011) and Altmann et al. (2012), the *base calling* step is where the images captured during the sequencing process of the newly generated strands are evaluated to produce sequence reads. In this evaluation, base-calling algorithms infer the actual nucleotide information from the acquired fluorescence-intensity data for each cluster of DNA templates, assigning a measure of uncertainty (or quality score) related to each base call. The reads are typically provided in a flat file format such as FASTQ, which has emerged as a *de facto* format for storing raw NGS read data (Cock et al., 2009). In this file format, single ASCII (American Standard Code for Information Interchange) characters are used to encode base quality as Phred-like quality

scores — a log-based measure of error probability, inherited from the Sanger sequencing era (Ewing et al., 1998; Ewing and Green, 1998; Cock et al., 2009; Pop, 2009), which expresses the probability of the base call being wrong (Equation 1.1) (Altmann et al., 2012):

$$Q_{phred} = -10 \times \log_{10} P(error) \tag{1.1}$$

Table 1.2 shows examples of some Phred quality scores.

Table 1.2: Examples of Phred quality scores.

Phred quality score	Probability of incorrect base call	Accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.90%
40	1 in 10000	99.99%

The base calling step is usually automatically performed by the NGS sequencing platform itself (Altmann et al., 2012) and its precise nature varies accordingly with the type of technology/vendor. In spite of continual improvement of NGS platforms' accuracy over time, problems with the raw data are common, e.g. contamination with adapters or fragments thereof, low quality ($< Q_{phred}$ 20) and ambiguous bases, chimeric reads, platform-specific artefacts, homopolymer stretches of incorrect length, and contaminating host nucleic acids (Meacham et al., 2011; Nakamura et al., 2011; Victoria Wang et al., 2012; Shao et al., 2013; Lu

et al., 2014). Although algorithms have been proposed to improve sequence quality over the manufacturers' algorithms — BayesCall (Kao et al., 2009; Nielsen et al., 2011), Ibis (Kircher et al., 2009; Altmann et al., 2012), and naiveBayesCall (Kao and Song, 2011; Altmann et al., 2012) for Illumina; PyroBayes (Quinlan et al., 2008; Altmann et al., 2012) for 454; and Rsolid (Wu et al., 2010; Altmann et al., 2012) for SOLiD™, among other examples — most users still rely entirely on the base calling output from the sequencing platforms (Nielsen et al., 2011; Altmann et al., 2012). Either way, reducing the error rate of base calls and improving the accuracy of the per-base quality score has important implications for assembly, polymorphism detection, and downstream population-genomic analyses (Nielsen et al., 2011). Thus, the base calling step is typically followed by data quality control which usually involves quality plots of the raw data, trimming/filtering steps, and the removal of contaminants (Lu et al., 2014). For the assessment of quality scores at each sequence position, software like SolexaQA (Cox et al., 2010; Altmann et al., 2012), FASTX-Toolkit (Hannon, 2009; Lu et al., 2014), and FastQC (Andrews, 2010; Altmann et al., 2012; Lu et al., 2014) are common choices. The information gathered can then be used to spot possible problems with the sample preparation or the sequencing run.

In reads produced by Illumina platforms, for example, base quality typically drops off sharply towards the end of read and trimming procedures are carried out

as a countermeasure. Both the mentioned SolexaQA and FASTX-Toolkit solutions provide specific modules tailored for the task, but other tools are also available: Sickle (Joshi and Fass, 2011; Lu et al., 2014), Scythe (Buffalo, 2011; Lu et al., 2014), Trimmomatic (Bolger et al., 2014), etc. For the removal of adapters and other contaminants, Cutadapt (Martin, 2011; Lu et al., 2014) can also be used. In general, the dataset types and further requirements of downstream analysis ultimately determine which programs are to be used (Lu et al., 2014).

1.2.3 Variant calling pipeline – *De novo* assembly stage for non-model organisms

Most SNP calling applications take a reference-based mapping approach, where mapped reads are 'compared' with the reference a base at a time, so genetic variants may be detected (Nielsen et al., 2011; Leggett and MacLean, 2014). This makes it difficult to apply such application/algorithms in non-model species, since a suitable reference genome is usually not available (Dou et al., 2012). A possible alternative is the *de novo* assembly of the reference sequence (Pop, 2009; Flicek and Birney, 2009; Imelfort et al., 2009; Paszkiewicz and Studholme, 2010), a complex, computationally intensive procedure (Paszkiewicz and Studholme, 2010; Baker, 2012; Henson et al., 2012; Clevenger et al., 2015).

Generally, with NGS methods, reads are either aligned to a reference genome or assembled *de novo*, so the data can be interpreted in a biologically meaningful

manner (Flicek and Birney, 2009; Nielsen et al., 2011). One assembly method uses the sequence of a closely related organism previously sequenced to characterise a newly sequenced one and is referred as *comparative assembly* (Pop, 2009; Bao et al., 2011; Nielsen et al., 2011). The second approach, *de novo* assembly, aims to reconstruct genomes that have never been sequenced (Pop, 2009; Bao et al., 2011), which is often the case of non-model organisms (e.g. crop plants). *De novo* genome sequence assembly is thus important for both generating new sequence assemblies for previously uncharacterised genomes — essential for cataloguing Earth's biological diversity (Pop, 2009) — as well as for identifying the sequence of individuals in a reference-unbiased way (Simpson and Durbin, 2012). Nevertheless, the two assembly approaches are not mutually exclusive, as some genome sequencing projects may combine the results of both (Pop, 2009; Paszkiewicz and Studholme, 2010).

'Assembly' is necessary as there is currently no technology capable of reconstructing the entire length of a DNA molecule; instead, the available technologies all produce smaller fragments or chunks (Schatz et al., 2010). Assembly was first introduced in the late 1970s (Staden, 1979), combining the products of shotgun sequencing and computation (Pop, 2009). It has been used since then to organise sequence fragments into the longer sequences they originate from (Bao et al., 2011). The fundamental concept is to group sequence chunks

(reads) into long contiguous sequences ('contigs') and to, then, where possible, group these into longer 'scaffolds', in order to reconstruct the original sequence (Bao et al., 2011). The process is often likened to solving a large jigsaw puzzle without knowledge of the final picture (Pop and Salzberg, 2008; Pop, 2009). This means that, once a new genome assembly is completed, it is difficult to establish what portions of it are real, missing or artefactual (Baker, 2012).

Mathematically, the *de novo* assembly problem is very difficult, irrespective of the sequencing technology used, falling in the class of NP-hard problems (Pop and Salzberg, 2008; Bao et al., 2011; Baker, 2012), which are "computational problems for which no efficient solution is known" (Pop and Salzberg, 2008). Particular challenges are, for example:

- genomic repeats of DNA that occur in near-identical form throughout a genome significantly complicate the assembly — multiple copies of these may collapse into a single sequence in the assembly, which constitutes misassembly (Salzberg and Yorke, 2005; Paszkiewicz and Studholme, 2010; Schatz et al., 2010; Henson et al., 2012) (Figure 1.8) —, in particular if they are longer than the length of a read (Pop, 2009);

- the complexity of the assembly increases dramatically with the number of reads being assembled, which is the case, for instance, for large genomes and/or shorter reads (Pop, 2009);

- real biological sequence data is not error-free, implying that many of the sequence reads contain mismatches with respect to the final sequence, which makes difficult the resolution of the latter by the assembly algorithm (Paszkiewicz and Studholme, 2010; Baker, 2012);

- additional factors, like heterozygosity and ploidy (Figure 1.9), polymorphisms (both at the level of single base changes and of small indel and larger structural variations), missing data due to lower quality or coverage — the total number of reads traversing a given locus — in difficult-to-sequence genomic regions, and high percentage of guanine and cytosine content for the targeted specific organism.

All of the above can potentially restrict the length of the contigs assembled and lead to gaps between these (Flicek and Birney, 2009; Baker, 2012; Hamilton and Buell, 2012; Schatz et al., 2012).

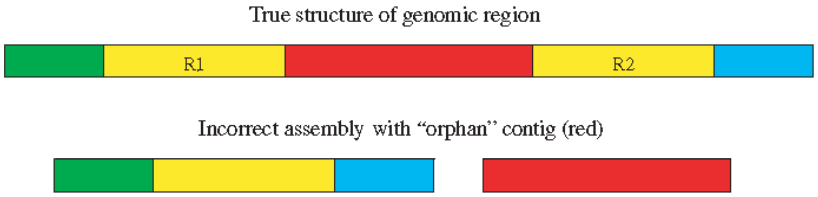

Figure 1.8: Repetitive sequences of the genome can complicate assemblies. As explained by Salzberg and Yorke (2005), assemblies can collapse around repetitive sequences: regions R1 and R2, in yellow, represent near-identical copies of the same DNA sequence separated by a unique region shown in red. If R1 and R2 are longer than the sequence read length available, then the assembler will not have any individual reads containing the entire repeat and its unique flanking sequences (the green and blue regions). This can result in a genome assembly like the one shown in the bottom part of the figure, with a contiguous stretch of DNA (a contig) comprising only one copy of the repeat, and incorrectly connecting the blue and green regions. Due to this, the red region won't connect to anything and will end up as a separate (potentially very short) contig. Adapted from Figure 1, Salzberg and Yorke (2005), "Beware of mis-assembled genomes", Bioinformatics, 2005, Volume 21, Issue 24, Pp. 4320-4321, by permission of Oxford University Press.

The introduction of NGS technologies has made this scenario much more complex (Baker, 2012), as data generation involves millions or even billions of much shorter reads (Thudi et al., 2012; Lu et al., 2014) than those produced by the traditional Sanger method. For example, as reads get shorter, coverage — formally defined as the average number of times each nucleotide is sequenced given a certain number of reads of a given length and the assumption that reads are randomly distributed across an idealised genome (Sims et al., 2014) — needs to increase to compensate for the inherent decreased connectivity (Schatz et al., 2010; Zhang et al., 2011). Apart from the read length and throughput aspects, error profiles of the new technologies also vary (Miller et al., 2010), with some of them

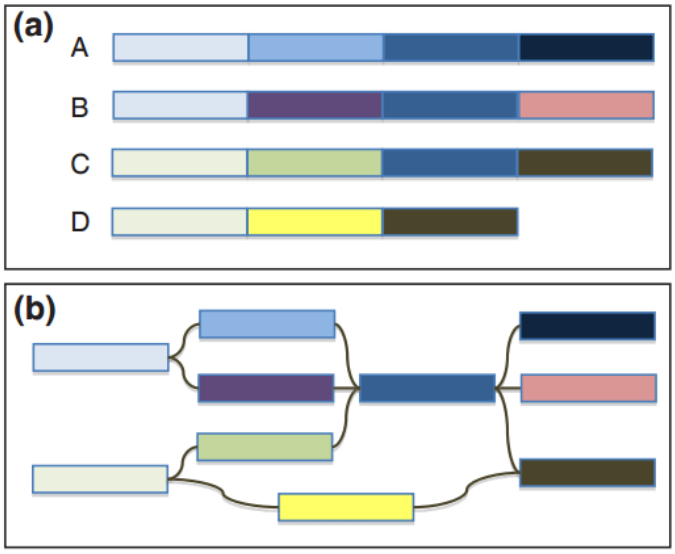

Figure 1.9: Ploidy, heterozygosity and their impact on the assembly, as explained by Schatz et al. (2012). (a) Schematic representation of a tetraploid genome, consisting of four chromosomes A to D with homozygosity/heterozygosity shown as different coloured blocks. (b) Even in the absence of repeats or sequencing errors, the assembly graph of the homozygous and heterozygous segments of the genome branches in a complex pattern, making the reconstruction of chromosomes into individual sequences extremely difficult. Adapted from Figure 2 (Schatz et al., 2012).

being platform-specific. For example, 454 system can 'stutter' in homopolymer regions (Pop, 2009; Gilles et al., 2011 in Henson et al., 2012); base substitutions are the most common error type for the Illumina (Bentley et al., 2008; Dohm et al., 2008 in Henson et al., 2012) and ABI SOLiD^{TM} (Metzker, 2010) platforms, with a higher proportion of errors occurring in the former when the previously incorporated nucleotide is a 'G' base (Dohm et al., 2008). Similar to Illumina, SOLiD^{TM} data has also shown under-representation of AT- and GC-rich regions (Harismendy et al., 2009).

All of the above pose new computational challenges to the assembly algorithms typically used in Sanger sequencing era (Huang and Madan, 1999; Adams et al., 2000; Myers, 1995; Myers et al., 2000; Waterston et al., 2002), like, for instance, the *Overlap-Layout-Consensus (OLC)* approach, in which fragments are overlapped so that shared identical regions between them are aligned. Assemblers designed to deal with Sanger reads were found to be impractical when dealing with NGS data (Henson et al., 2012) especially regarding aspects like computer memory footprint and processing speed.

Due to these challenges, since 2005, several assembly software packages have been created or revised specifically for *de novo* assembly of next-generation sequencing data (Miller et al., 2010): ABySS (Simpson et al., 2009), ALLPATHS (Butler et al., 2008), CABOG (Miller et al., 2008), Edena (Hernandez et al.,

2008), EULER-SR (Chaisson and Pevzner, 2008), MaSuRCA (Zimin et al., 2013), Newbler (Margulies et al., 2005), QSRA (Bryant et al., 2009), SHARCGS (Dohm et al., 2007), SOAPdenovo (Li et al., 2009d), SSAKE (Warren et al., 2007), VCAKE (Jeck et al., 2007), and Velvet (Zerbino and Birney, 2008) are some examples, but many others are available (Wikipedia, 2005).

De novo assembly strategies employed by the exemplified tools vary between 'greedy', OLC, and Eulerian (Pop and Salzberg, 2008) or de Bruijn graph (Bao et al., 2011), but others, like the string graph, are also available (Henson et al., 2012; Simpson and Durbin, 2012). However, most popular short read assemblers rely on the de Bruijn graph model, which requires breaking the reads up into sequences (or substrings) of a fixed length k, called k-mers (Henson et al., 2012; Simpson and Durbin, 2012; Clevenger et al., 2015). Some *de novo* transcriptome assemblers, like Trinity (Grabherr et al., 2011; Haas et al., 2013) and Oases (Schulz et al., 2012), also use this approach (Martin and Wang, 2011).

As detailed in works like Henson et al. (2012), Leggett and MacLean (2014), and Langmead (2014), instead of storing information about reads and overlaps explicitly, a de Bruijn graph is a data structure made up of nodes representing all k-mers that appear in reads, linked by edges between pairs of k-mers that appear consecutively (overlap by k-1 nucleotide; Figure 1.10). A read whose k-mers are all contained in other reads adds nothing to the graph, and so memory

requirements scale well with the coverage burden imposed by NGS (Henson et al., 2012). In theory, in the absence of read errors and with a k 'word' long enough to encompass the longest repeat in a single k-mer, the genome then corresponds to a path through the graph (Henson et al., 2012; Leggett and MacLean, 2014). However, sequencing data do contain errors and carry information representing true genetic variation (Leggett and MacLean, 2014). Furthermore, repetitive sequences are usually longer than the read length available (let alone the k length itself) (Leggett and MacLean, 2014). Thus, when the graph is built, sequence errors can cause dead ends ('tips') in the path, ambiguities like polymorphisms generate 'bubbles', and genomic repeats may become cycles (Leggett and MacLean, 2014). Unambiguous contigs are represented by non-branching paths, while the ambiguities at the boundaries of repeats are explicitly represented in the graph as branch nodes ('constrictions') (Henson et al., 2012).

After the graph is constructed, it is simplified and contigs are extracted from it. Typically, non-branching paths of k-mers are merged into one node, thereby saving further space (Henson et al., 2012). Scaffolding and gap closure can proceed after unambiguous contigs are found.

Due to the reduced computational effort and greater efficiency obtained, de Bruijn graphs have proven to be of great utility as the underlying data model which almost all *de novo* assembly algorithms designed to use short read data

have been implemented (Leggett and MacLean, 2014).

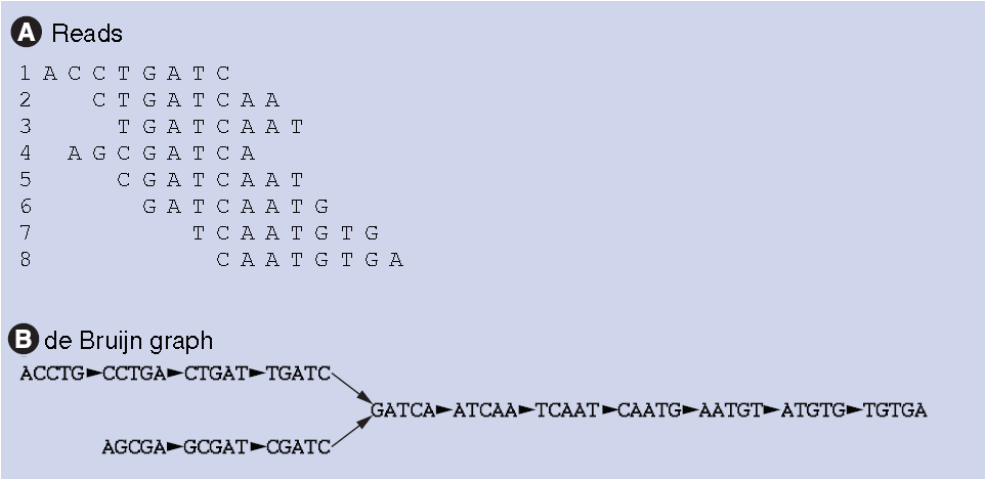

Figure 1.10: de Bruijn graph structure for assembly, as per Henson et al. (2012). (A) Eight reads aligned are shown. (B) The de Bruijn graph, in which nodes are k-mers and edges indicate that some read contains two k-mers consecutively. Reads such as number two add nothing to the de Bruijn graph. Adapted from Figure 2 (Henson et al., 2012). Reproduced with permission of FUTURE MEDICINE LTD., from "Next-generation sequencing and large genome assemblies", Henson et al. (2012), Pharmacogenomics, Volume 13, Issue 8, p. 906, Copyright 2012; permission conveyed through Copyright Clearance Center, Inc.

De novo genome (and transcriptome) assemblies based exclusively on NGS data are now feasible, of at least good draft quality, either for simpler genomes, like the extremophile crucifer *Thellungiella parvula* (Dassanayake et al., 2011 in Hamilton and Buell, 2012), or large mammalian genomes, like the giant panda (Li et al., 2010 in Paszkiewicz and Studholme, 2010), albeit with limitations (Paszkiewicz and Studholme, 2010; Alkan et al., 2011; Birney, 2011).

A good reference sequence is crucial for the conventional mapping-based SNP calling application in non-model organisms. Thus, many short read-related *de novo*

sequencing projects still rely on a hybrid approach with the inclusion of, at least, some longer sequence reads source (Paszkiewicz and Studholme, 2010; Hamilton and Buell, 2012; Kumar et al., 2012; Leggett and MacLean, 2014). This is due to the highly repetitive nature of some genomes, as often is the case in plants. As mentioned earlier, repeating sequences of DNA confound *de novo* assembly approaches, a problem exacerbated by short read length since fewer repeats can be resolved (Henson et al., 2012). Just increasing read coverage alone is not sufficient for resolving repeats accurately (Schatz et al., 2010; Henson et al., 2012). Instead, paired reads (mate-pair or paired-end sequences) — consisting of two reads generated from a single fragment of DNA and separated by a known distance as long as the pair separation distance is longer than the repeat (Schatz et al., 2010) —, are usually applied to help with repeat resolution and the ordering of the contigs along the genome (Pop and Salzberg, 2008; Flicek and Birney, 2009; Hamilton and Buell, 2012; Henson et al., 2012). To obtain better assembly results, it is important to also consider the use of libraries with different insert sizes to facilitate scaffolding of underlying contigs (Paszkiewicz and Studholme, 2010; Hamilton and Buell, 2012; Henson et al., 2012). Thus, appropriate experimental design can help overcome some of the problems inherent to NGS technologies. Depending on the project and biological question, it is still also possible to opt for a reduced-representation approach instead of whole-genome sequencing (Imelfort et al., 2009). SNPs can

still be identified using transcriptome or reduced-representation data (Davey et al., 2011; Lai et al., 2012).

1.2.4 Variant calling pipeline – Alignment stage

In many types of genomic analysis, alignment or mapping of NGS short reads to a reference sequence is the first step (Flicek and Birney, 2009; Horner et al., 2010; Bao et al., 2011; Ruffalo et al., 2011; Altmann et al., 2012; Fonseca et al., 2012). In SNP calling, sequence reads can be mapped back to a reference sequence, if available, and SNPs are then called based on one or multiple samples, which is usually accompanied by a genotyping stage (Nielsen et al., 2011; Dou et al., 2012; Clevenger et al., 2015).

The high throughput makes NGS technologies particularly suitable for genetic variation studies of sizeable cohorts of individuals with a known reference (Metzker, 2005; Bentley, 2006; Li et al., 2009b; Liao and Lee, 2010). Resequencing, which is the most common application for NGS (Ratan et al., 2010), has been used to identify, for instance, large numbers of SNPs in plant genomes, such as Arabidopsis, rice, soybean and maize (Lai et al., 2012), but also in other kingdoms (Treangen and Salzberg, 2012; Cantarel et al., 2014).

The primary challenge of the NGS alignment used in resequencing is, given a large set of short reads from an individual's genome, to efficiently find the true location of each read in a potentially extensive reference sequence. Additionally,

the alignment procedure must be capable of distinguishing between technical sequencing errors and true differences between the donor and reference genomes as well as dealing with the repeats of the latter (Ruffalo et al., 2011, 2012; Fonseca et al., 2012). Smolka et al. (2015), for instance, state that "mapping reads to a genome remains challenging, especially for non-model organisms with lower quality assemblies, or for organisms with higher mutation rates". Factors like genetic variation, sequencing error, short NGS read length, different read lengths used, the large volume of reads to be mapped, quality of the reference genome, and reference sequence complexity (such as GC content and repetitive regions) can complicate the task (Ruffalo et al., 2011; Smolka et al., 2015). Alignments are also affected by the genetic distance between the reference individual and newly sequenced genomes. Moreover, according to Sims et al. (2014), even the best mapping algorithms cannot align all reads to the reference sequence. The authors cite structural rearrangements or insertions in the query genome, or deletions in the reference, as additional causes for that. They also reinforce that it is not possible to unambiguously assign reads to all genomic regions, as some of these will contain low-degeneracy repeats or low-complexity sequences.

Repetitive regions, for example, represent technical challenges not only to *de novo* assemblies but also for alignments, due to the ambiguities they provoke from a computational perspective (Treangen and Salzberg, 2012) (Figure 1.11).

Multi-mapped reads are the product of such ambiguities and are characterised as those that align to multiple locations, with similar alignment scores, due to either originating from repetitive regions and/or due to their short length (Fonseca et al., 2012).

Figure 1.11: Ambiguity in read mapping, as explained by Treangen and Salzberg (2012). The read shown maps to two locations, (a) and (b). In (a), there is a mismatch. In (b), a deletion. If mismatches are less penalised than a gap by the mapping algorithm (e.g. if it assumes that substitutions are more likely than deletions), the program will assign the read to location (a). However, the source (donor) DNA might have a genuine deletion in location (b), meaning that this is the true position of the read. Adapted from Figure 1b (Treangen and Salzberg, 2012), by permission from Macmillan Publishers Ltd: Nature Reviews Genetics, copyright 2012.

Many mappers (or aligners) are available to deal with the task of trying to find the correct locations of NGS short reads in relation to a given reference sequence (Ruffalo et al., 2011; Altmann et al., 2012; Clevenger et al., 2015) — nearly one hundred, by mid-2015, accordingly to Smolka et al. (2015). The following are examples: BFAST (Homer et al., 2009), BLAT (Kent, 2002), Bowtie (Langmead et al., 2009), Bowtie2 (Langmead and Salzberg, 2012), BWA (Li and Durbin, 2009), ELAND (Cox, 2007), MapNext (Bao et al., 2009), MAQ (Li et al., 2008a), Mosaik (Lee et al., 2014), Novoalign (Novocraft, 2008), PerM (Chen et al., 2009), RMAP (Smith et al., 2008), SHRiMP (Rumble et al., 2009), SeqMap (Jiang and Wong, 2008), SOAP/SOAP2 (Li et al., 2008b, 2009c), SOCS (Ondov et al., 2008),

SSAHA/SSAHA2 (Ning et al., 2001), Stampy (Lunter and Goodson, 2011), and ZOOM (Lin et al., 2008). Many others can also be tracked (EMBL-EBI, 2012; Fonseca et al., 2012; Wikipedia, 2008).

Two distinct computational techniques are commonly used among the different algorithms: the Burrows-Wheeler transform (BWT) (Burrows and Wheeler, 1994 in Altmann et al., 2012), for efficient data compression; and hash table data structure or *'indexing'*, which accelerates the alignment step by either hashing the reads or the reference sequence (Nielsen et al., 2011; Altmann et al., 2012; Shang et al., 2014). Figure 1.12 illustrates these algorithmic approaches.

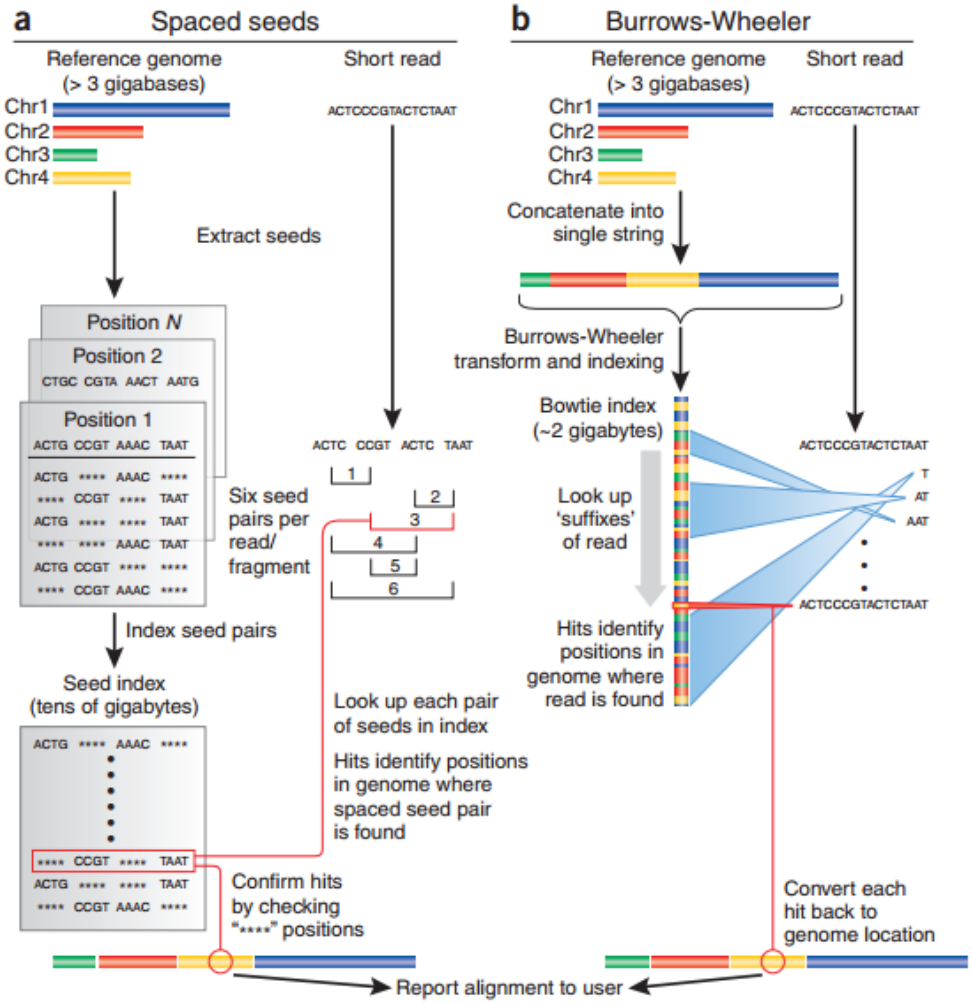

Figure 1.12: Examples of algorithmic approaches to tackle the NGS short-read mapping problem, as explained by Trapnell and Salzberg (2009): (a) Algorithms such as MAQ (Li et al., 2008a) are based on 'spaced-seed' indexing. In this approach, each position in the reference is cut into equal-sized pieces, the 'seeds', and these seeds are paired and stored in a lookup table. Each read is also cut up following the same scheme and pairs of seeds are used as keys to look up matching positions in the reference. (b) Algorithms like Bowtie (Langmead et al., 2009) rely on the Burrows-Wheeler transform concept, storing a memory-efficient representation of the reference genome. Reads are aligned character by character, from right to left, against the transformed string. Blue 'beams' in the figure indicate interval updates in the transformed string with each new character evaluated. After the processing of all characters in the read, alignments are represented by any positions within the interval. Adapted from Figure 1 (Trapnell and Salzberg, 2009), by permission from Macmillan Publishers Ltd: Nature Biotechnology, copyright 2009.

As an addendum, it is important to highlight that short-read mappers are typically employed to solve one 'version' of the alignment problem, in which reads must be aligned without allowing large gaps. A variation of the problem arises primarily in RNA-Seq, in which alignments are allowed to have large gaps (corresponding to introns) and where the task is undertaken by tools that fall into the category of spliced-read mappers (Trapnell and Salzberg, 2009).

1.2.5 Variant calling pipeline – Post-alignment stage

The post-alignment stage is one of the steps shared by nearly all HTS applications (Altmann et al., 2012). Alignments are stored in the sequence alignment/map (SAM) format (Li et al., 2009a), which can be converted into its binary (BAM) equivalent (Clevenger et al., 2015). Normally, these files containing the mapped reads undergo some transformations before they can be used in downstream analysis. As highlighted in the work of Anders et al. (2013), the SAMtools suite (Li et al., 2009a) is used to prepare variations of the mapped reads, like a sorted and indexed version of the BAM file, which can be used in genome browsers and visualisation tools (e.g. IGV (Robinson et al., 2011) and Tablet (Milne et al., 2010)). The Picard suite (Broad Institute, 2014a) is another option for sorting alignments prior to the variant calling stage (Altmann et al., 2012).

Additional post-alignment measures have been introduced to improve data

quality prior to the variant calling stage. For example, as mentioned earlier in this chapter, NGS platforms, like Illumina, produce amplification biases introduced by the PCR process. Due to this, artefacts, i.e. reads or read pairs starting at exactly the same position and having the same insert length, respectively, may be introduced. For calculating genotype-likelihoods in contemporary approaches of the variant calling stage, there is an implicit assumption of independence among reads (Nielsen et al., 2011). PCR duplicates violate this assumption since they are non-independent measurements of a given sequence (Broad Institute, 2015a). It is therefore recommended practice to have these marked or removed (Altmann et al., 2012). Again, SAMtools and Picard provide the means for solving this task (Figure 1.13).

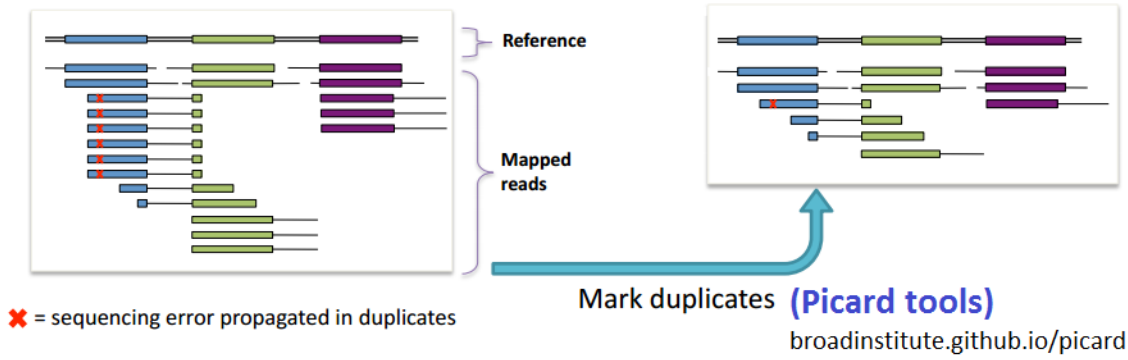

Figure 1.13: PCR duplicates and an example strategy for remediation. Duplicates violate assumptions of non-independent observations, so usually the "best" copy of the read is kept to mitigate the effects of errors. Adapted from Broad Institute (2015b).

Other measures, typically established in the shape of best practices pipelines (gkno, 2013; Broad Institute, 2015a; bcbio-nextgen, 2015) may include removal

of (if present) all non-unique alignments — reads with more than one optimal alignment denoting cases where the true origin of a read cannot be determined —, realignments around putative indels (to prevent artificial SNPs), and base quality score recalibration (since Phred-like quality scores issued by sequencing platforms can co-vary with both the base position in the read and the identity of neighbouring bases) (Li et al., 2009b; McKenna et al., 2010; Nielsen et al., 2011; Altmann et al., 2012; Farrer et al., 2013; Cornish and Guda, 2015).

1.2.6 Variant calling pipeline – SNP calling stage

Having mapped and post-processed the reads, the next step in the computational pipeline is to call SNPs (Nielsen et al., 2011; Altmann et al., 2012; Clevenger et al., 2015) using programs like FreeBayes (Garrison and Marth, 2012), GATK (DePristo et al., 2011), MAQ (Li et al., 2008a), SAMtools (Li et al., 2009a), SOAPsnp (Li et al., 2009b), Sniper (Simola and Kim, 2011), and VarScan (Koboldt et al., 2009).

In theory, a SNP is identified when a nucleotide from an accession read differs from the reference genome at the same nucleotide position (Kumar et al., 2012). Independently of the alignment accuracy and post-alignment measures, errors can go unnoticed, thereby propagating into the SNP discovery stage (Farrer et al., 2013). So, the following are some of the direct (and indirect) challenges posed at this stage:

- Systematic (Meacham et al., 2011; Nakamura et al., 2011) and random read errors;
- Low coverage (for instance, apart from the technical method employed towards an aimed depth, achieved coverage can be limited by the inherent random nature of the sampling of the sequencing as well as may vary accordingly to the nature of the targeted sequencing application; e.g. genome, transcriptome, or exome) (Sims et al., 2014);
- Ploidy (for instance, SNP identification in polyploids is more challenging due to the need for distinguishing homeologous SNPs from allelic ones) (Clevenger et al., 2015);
- Genetic variations (e.g. copy number variation, insertion, deletion, inversion, and rearrangements) (Yu and Sun, 2013);
- Repetitive genomic regions (e.g. interspersed repetitive elements and paralogous genes present in eukaryotic genomes) (Simola and Kim, 2011).

Furthermore, as Single Nucleotide Variant (SNV) detection occurs with individual base pair resolution, any sequencing error can potentially lead to an incorrect SNP call (Yu and Sun, 2013). Thus, as emphasized in the review of Nielsen et al. (2011), under these circumstances, accurate SNP and genotyping calling are both difficult to achieve: there is often a substantial amount of

uncertainty associated with the results obtained which needs to be quantified accurately, as it influences downstream analyses based on the putative SNPs and genotypes.

Earlier SNP and genotype calling approaches, which worked well at high sequencing depths (>20x), were mainly based on counting the abundance of high-quality nucleotide alleles at a given locus and application of fixed cutoff rules for making a call (Bentley, 2006; Wang et al., 2008; Nielsen et al., 2011; Altmann et al., 2012). In recent years, extensive research of the subject has taken place to reduce and quantify this uncertainty with sophisticated algorithms. Such solutions integrate several sources of information (e.g. sequence and alignment quality metrics) and rely on some probabilistic framework to generate likelihoods (Nielsen et al., 2011; Leggett and MacLean, 2014). Depending on the number of samples and the depth of coverage, either a single- or multi-sample calling procedure may be carried out (Nielsen et al., 2011). The likelihoods are coupled with prior known information about SNPs/genotypes — databases like dbSNP (Sherry et al., 2001 in Altmann et al., 2012) or from SNP calling in multiple individuals (Altmann et al., 2012) — and usually some sort of statistically meaningful 'quality score' for the final calls is generated. Probabilistic methods also work well for moderate or low sequencing depths (<5x per site per individual, on average) by being more robust to avoid under-calling of heterozygous genotypes (Nielsen et al., 2011).

Most modern variant calling tools call genotypes of the samples involved in the mapping in conjunction with the SNP calling task, as part of the same — usually Bayesian — model (Nielsen et al., 2011; Liu et al., 2013; Yu and Sun, 2013; Clevenger et al., 2015). Yu and Sun (2013), for instance, cite SOAPsnp (Li et al., 2009b), SAMtools (Li et al., 2009a), and Unified Genotyper (UGT) in GATK (McKenna et al., 2010; DePristo et al., 2011) as examples, stating that such approaches compute the posterior probability for each possible genotype, choosing the one with the highest probability as the consensus genotype. A SNP is called at a given position if its consensus genotype is different from the reference.

Figure 1.14 illustrates the idea of a typical probabilistic method. Additionally, patterns of Linkage Disequilibrium (LD) — the non-random association between alleles at different loci (Pabinger et al., 2014) — and allele frequencies information can be also incorporated within the probabilistic framework to improve the SNP/ genotype calling results (Nielsen et al., 2011; Altmann et al., 2012). As stated by Nielsen et al. (2011), "several different population genetic methods have been developed for *imputation* — the use of a set of reference haplotypes to infer an individual's genotype when data are missing or incomplete — of missing data in SNP data sets".

After the SNP calling procedure, usually some filtering is necessary, based on adequate criteria for the species of interest and experimental design, to reduce

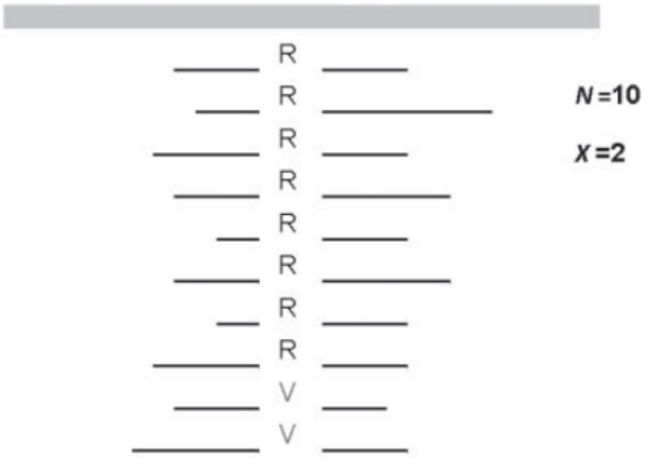

Figure 1.14: Example of a probabilistic method in single nucleotide variant calling from NGS data, as explained by Martin et al. (2010). Schematic of 10 aligned NGS reads (R = reference nucleotide; V = variant nucleotide) for a single base locus. N is the read depth while X is the variant count. The authors describe a possible generic probability model as follows: for a diploid individual (i), a specific bi-allelic base position is sampled at random N_i times from a large pool of sequences. X_i copies of variant nucleotide V and N_i - X_i copies of reference nucleotide R are observed. The probabilities that V is falsely called R and vice-versa are equal and the probability of this error is denoted α. G_i is the true genotype of the individual. In a Bayesian approach, depending on prior genotype frequencies (for VV and RV in this context), given sequence data $\{N_i,X_i\}$, and the nucleotide-read error rate (α), the genotype with maximum posterior probability is assigned to an individual. So, for the scenario shown, this could mean a heterozygous SNV call (genotype RV predicted), V nucleotides potentially being classified as errors and no variant called (genotype RR predicted), or, alternatively, R nucleotides being classified as errors and a homozygous SNV called (genotype VV predicted). Adapted from Wikipedia (2014) and Figure 1, Martin et al. (2010), "SeqEM: an adaptive genotype-calling approach for next-generation sequencing studies", Bioinformatics, 2010, Volume 26, Issue 22, Pp. 2803-2810, by permission of Oxford University Press.

the number of FP SNP calls and to strike a balance between sensitivity and specificity (Clevenger et al., 2015). Commonly applied filters check for minimum and maximum read depth, adjacency to indels, strand bias, quality score, read mapping quality, base quality, minor allele frequency, etc. (Li et al., 2008a; Nielsen et al., 2011; Altmann et al., 2012; Cantarel et al., 2014; Clevenger et al., 2015).

The usual output of the variant calling tools is a VCF (Variant Call Format) file (Danecek et al., 2011b; SAMtools Project, 2015), a file format developed for the 1000 Genomes Project. As claimed by its developers, in the same manner that SAM/BAM format was specified to standardise the storing of NGS read alignments-related information, the VCF file was proposed for storing the most common types of sequence variation information (e.g. SNPs, indels, structural variants, etc.), in order to improve the interoperability between the stages of the NGS variant calling workflow (Danecek et al., 2011a). Thus, filtering may be carried out by tools like SAMtools (via the script *'vcfutils.pl'*) and VCFtools (Danecek et al., 2011b), directly over such kind of file.

For making sense of SNP data, since usually a massive number of candidate variants is generated and remains after the filtering step, tools for automated variant annotation may be applied (e.g. ANNOVAR (Wang et al., 2010; Altmann et al., 2012), SnpEff (Cingolani et al., 2012), and others found at G2P (2010)) (Altmann et al., 2012). Such tools are designed to quickly process large numbers

of called variants, are available for different species, can be integrated with variant calling pipelines (e.g. GATK), and aim to predict the coding effects of genetic variations in whole genome sequences (Cingolani et al., 2012; Altmann et al., 2012).

SNP validation is also a task typically performed in this stage by using a subset of the identified candidates. For this, medium- and high-throughput assays (e.g. competitive allele-specific PCR (KASP) (LGC Genomics, UK), high-resolution melting analysis (HRM), Infinium chips (Illumina, San Diego, CA), etc.) are options as well as Sanger sequencing for lower-throughput needs (Clevenger et al., 2015).

1.3 FP SNP examples

As explored throughout this chapter, falsely called SNPs can arise due to inherent challenges associated with NGS data generation (e.g. shorter reads) which induce errors that are propagated to the subsequent SNP/genotype calling stage (Nielsen et al., 2011; Altmann et al., 2012; Farrer et al., 2013; Li, 2014b). Apart from NGS-related sequencing errors, it is important to reinforce that most genomes — particularly eukaryotic — contain a significant portion of repetitive sequences, and this too contributes to FP SNPs (Dou et al., 2012). Here, some examples of errors which may generate different types of FP SNPs are provided along with their main characteristics.

1.3.1 FP SNPs due to sequencing errors and sequence-specific errors

It is known that base-call errors are more likely towards the ends of reads produced by NGS platforms (Dohm et al., 2008; Taub et al., 2010; Meacham et al., 2011). Read errors can look like variants and hence be falsely called as SNPs (Bravo and Irizarry, 2010; Taub et al., 2010) (Figure 1.15).

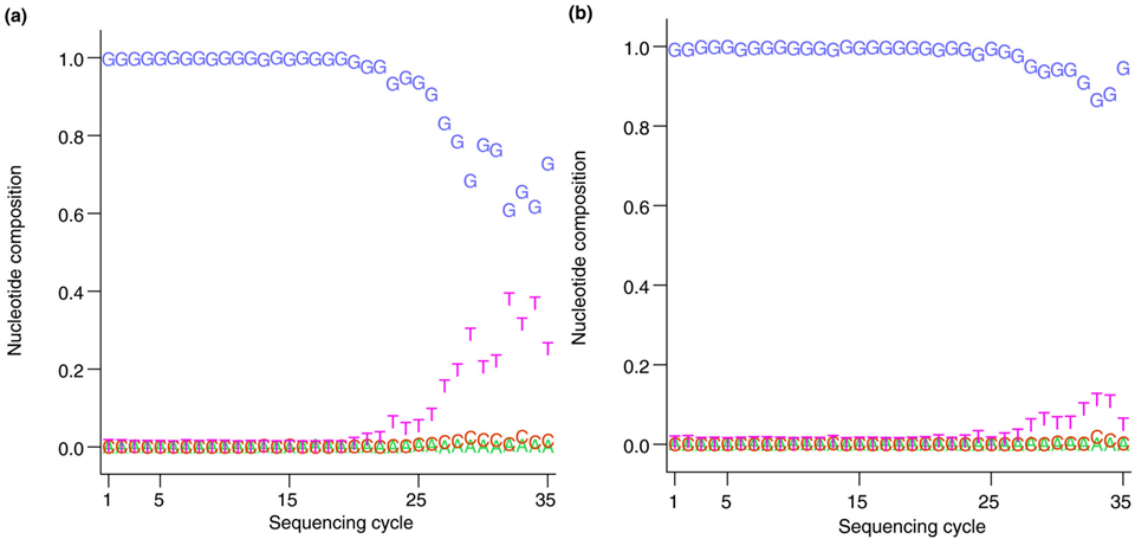

Figure 1.15: Example of base-calling error bias and improvement effect after countermeasure applied, as detailed by Taub et al. (2010). (a) Results obtained with a given default base-calling program. (b) Results obtained after application of base-calling correction by Bravo and Irizarry (2010). As per the authors' explanation, the x-axis shows read cycle and the coloured points indicate the percentage of calls at each cycle that were made for a particular nucleotide. In (a), 'T' bases become much more frequent in reads that align to the SNP site only at later sequencing cycles, indicating a technical bias in base-calls at this position. In (b), after improved base-calling, a strong reduction in this bias is observed and the location is no longer called as a variant by MAQ tool. Adapted from Figure 1 (Taub et al., 2010).

It is also known that surrounding sequence motifs (e.g. 'GG', 'GGC', 'GGT', etc.) influence error frequencies (Nakamura et al., 2011; Meacham et al., 2011);

these are referred to *sequence-specific errors* (SSEs). Taking the 'GGC' triplet as an example, errors are commonly found downstream of it, in the reads in forward direction, and upstream of the 'CCG' triplet, in the reads in the reverse direction, forming a characteristic visible triangular pattern when associated mappings are inspected (Nakamura et al., 2011). The presence of this kind of error may result in reference misassembly and subsequent FP SNPs.

Meacham and co-workers (2011) also report additional errors encountered at some genomic positions with greater frequency than can be explained by the previously mentioned phenomenon and refer to them as *systematic errors* (Figure 1.16). The authors explain that the main concern regarding systematic errors is that they may be incorrectly annotated as heterozygous sites in an individual or as rare variants in a population.

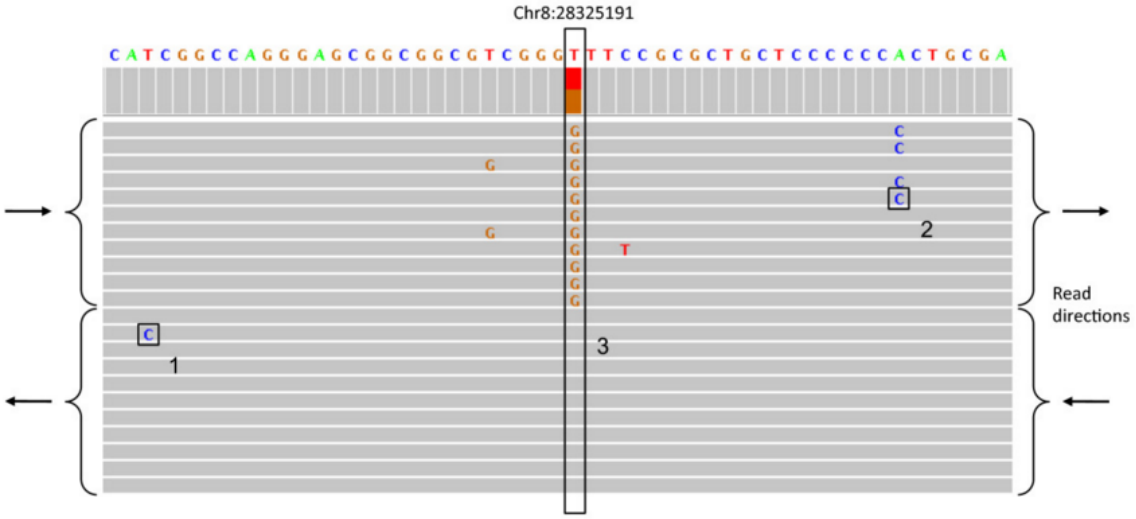

Figure 1.16: Types of errors, as detailed by Meacham et al. (2011). Screenshot from the IGV browser (Robinson et al., 2011) showing three types of errors in reads from an Illumina sequencer: (1) A random error most probably related to the *position* close to the end of the read. (2) Random error likely due to *sequence-specific error* — in this case, a sequence of 'Cs' is probably inducing errors at the end of a low complexity repeat. (3) *Systematic error* likely due to 'GGT' sequence motif and the previous 'GGC' motifs creating phasing problems. An incorrect SNP call occurs at the systematic error locus (coloured bar in top panel). Adapted from Figure 1 (Meacham et al., 2011).

1.3.2 FP SNPs due to duplicates

PCR duplicates, already explored in subsection 1.2.5, arise within the NGS library preparation and sequencing process, more specifically when, for instance, two copies of the same original molecule get onto different beads or different primer lawns in a flowcell (CureFFI.org, 2012). Optical duplicates are another kind of duplicated reads and are artefacts of the sequencing technology used by Illumina. They are sequences from one cluster that were erroneously identified by the software as representing multiple adjacent clusters (Whiteford et al., 2009) —

therefore occurring during the image detection of nucleotide incorporation. They appear as tightly packed clusters of identical sequences with identical start/end positions, similar to PCR duplicates.

In general, a read error in a set of duplicates (either PCR or optical) resembles a SNP with the reference (Figure 1.17). Another problem which may arise due to duplicates is as follows: if *de novo* assembly software is not able to distinguish between non-duplicate and duplicate reads and these latter constitute the vast majority, an erroneous base may be actually used to construct the consensus reference sequence. If, later, the duplicated reads with the 'erroneous' base are post-alignment removed, reads with the 'correct' base (of the true reference) will cause a FP SNP.

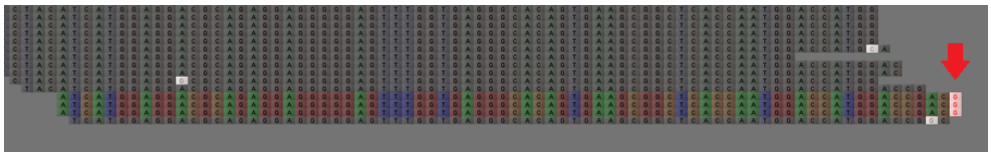

Figure 1.17: Optical duplicate and a resulting FP SNP. A read error in the duplicate reads looks like a SNP to the SNP discovery software, as seen in Tablet tool (Milne et al., 2010) screenshot (and highlighted by the red arrow). Adapted from M. Bayer (unpublished material).

1.3.3 FP SNPs due to reference misassembly

As exemplified in the previous topic, a single-nucleotide reference misassembly can be generated if an 'erroneous' base is used in an assembly leading to a potential FP SNP. The same kind of assembly computation problem can be caused by

different classes of very similar sequences, like Simple Sequence Repeats (SSRs) — repeats that either occur outwith genic regions or sometimes even within genes — and paralogs — pairs of genes that have arisen through gene duplication. Specifically regarding groups of paralogs, subsequent specialisation of function can give rise to 'gene families' (Golicz et al., 2014). Such examples of very similar but distinct sequences can act as confounding subjects to assemblers and, later, reads may get mismapped onto the misassembled locus (M. Bayer, unpublished material) (Figure 1.18).

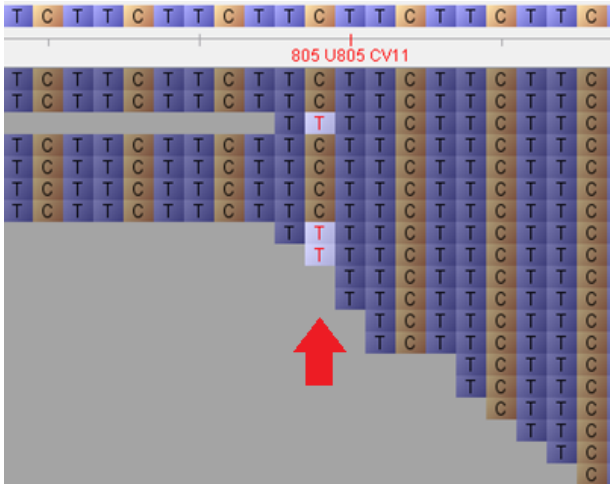

Figure 1.18: Reference misassembly and a resulting FP SNP. Screenshot of Tablet tool showing a heterozygous SNP (highlighted by the red arrow) due to incorrect fusion, during the assembly process, of two transcripts which contain the same kind of SSR. Adapted from M. Bayer (unpublished material).

1.3.4 FP SNPs due to read mismapping

Assembly and mapping issues like the ones already explored in this chapter (subsections 1.2.3 and 1.2.4) (e.g. misassembly and gaps provoked by collapsed

repeats; poor assembly resolution due to factors like sequencing errors, ploidy, polymorphisms, and heterozygosity; or ambiguity in read mapping due to repetitive regions in the reference sequence) may provoke FP SNPs due to read mismapping (e.g. missing references in the assembly). In general, wrongly aligned reads — without considering here the specific cause of the misalignment — may result in artificial discrepancies with the reference which, in turn, may falsely be classified as SNPs in the downstream processing (Nielsen et al., 2011; Altmann et al., 2012; Ruffalo et al., 2012; Farrer et al., 2013; Li, 2014b) (Figure 1.19).

Farrer et al. (2013), for instance, when introducing and benchmarking their BiSCaP SNP caller as well as other counterpart tools, demonstrated how alignment (and subsequent) SNP calling significantly varied, in general, in terms of FP SNPs obtained. In one of their tests with BiSCaP, for example, after having the reference sequence modified with random mutations *in silico*, 84 FP SNPs arose. Without the reference sequence alteration, none of these FP SNPs were observed and, as per the authors, this difference in numbers might have been caused by misaligned reads.

As explained by Li (2014b) in another investigation, when a read is mapped to the reference genome, the mapper chooses the optimal pairwise alignment for each read independent of the others. For multiple reads mapping to the same region, the combination of each respective optimal pairwise alignment does not always yield

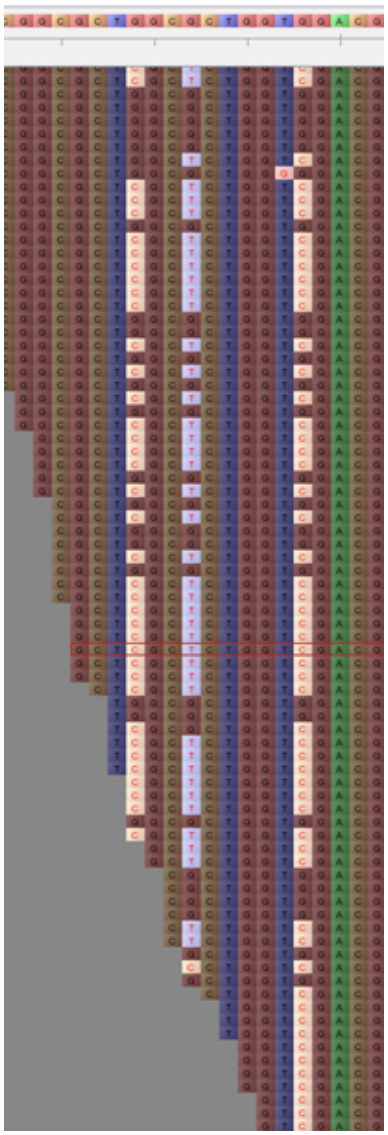

Figure 1.19: Read mismapping and resulting FP SNPs. Screenshot of Tablet tool showing heterozygous SNPs due to reads getting mapped to somewhere other than their true origin. Typically, several variants in phase occur together. Adapted from M. Bayer (unpublished material).

the optimal multi-alignment of all reads. Thus, if a variant caller relies on a given suboptimal multi-alignment, false variants or genotypes may be produced (Figure 1.20). According to the author's observations, even tools that rely heavily on realignment for both SNP and indel calling (e.g. FreeBayes (Garrison and Marth, 2012), HaplotypeCaller in GATK (McKenna et al., 2010; DePristo et al., 2011), and Platypus (Rimmer et al., 2014)), fail to generate the optimal realignment in low-complexity regions (LCRs) and, for most of the cases reviewed, local assembly with fermi (Li, 2014a) was suggested as a more effective countermeasure.

Figure 1.20: Example of misalignment, as explained by Li (2014b): the truth allele is derived from local assembly. Three read misalignments and their correct alignments are shown below it. Read 'errRead1' is aligned without gaps, as its 3' end is a substring of the 18 bp deletion. Read 'errRead2' is aligned with a 6 bp insertion, as this alignment is better than having two long deletions. Read 'errRead3' is aligned without gaps but with 7 mismatches. According to the study, except HaplotypeCaller, which locally assembled reads, other callers all called multiple heterozygotes around the region in question. Adapted from Figure 4, Li (2014b). Permission from "Toward better understanding of artifacts in variant calling from high-coverage samples", Bioinformatics, 2014, Volume 30, Issue 20, Pp. 2843-2851, by Oxford University Press.

The same study (Li, 2014b) also mentions about another experiment in which reads from a human haploid cell line were mapped to three different versions of the human genome: GRCh37, GRCh38, and hs37d5. As per the author's explanation,

the latter was used by the 1000 Genomes Project and contains extra 35.4 Mb *decoy* sequences — derived from *de novo* assemblies and that are supposed to attract many mismapped reads — and which are likely to be missing from the primary assembly GRCh37. After calling variants, the author observed twice as many false heterozygous calls from GRCh37 in comparison to hs37d5. This indicated that the decoy sequences indeed attracted many mismapped reads consequently improving the variant calling stage results. This is also an example which suggests that a more complete reference sequence can prevent mismapping of reads which, later, may generate FP SNPs.

Chapter 2

False positive SNP generation due to reference misassembly

Disclaimer

This chapter was based on a previous investigation carried out by A. Golicz (unpublished undergraduate honours project) in which a SNP calling pipeline algorithm was developed to classify FP SNPs in terms of their distinct patterns of occurrence. One such pattern was that of homozygous FP SNPs. The study presented here aimed to extend that exploration by implementing an automated way of quantifying and reporting whether this kind of event occurs due to a particular reference misassembly scenario, which, in its turn, may originate from closely related paralogs in the genome under verification.

2.1 Introduction

A previous mapping-based SNP calling investigation, designed to classify FP

SNPs in terms of their distinct patterns of occurrence (A. Golicz, unpublished undergraduate honours project), made a Trinity (Grabherr et al., 2011) *de novo* assembly of a wild barley (*Hordeum spontaneum*) sample (acession number WBDC016) whole transcriptome (RNA-Seq) and had the reads used in the assembly mapped back to the computed reference using the Bowtie 1 tool (Langmead et al., 2009). The dataset was composed of 76 bp long single-end reads obtained from an Illumina sequencing platform. The expectation of the study was that the only possible SNPs that could arise from using the same set of reads for the assembly of the reference sequence as well as for the mapping to the reference would present as heterozygous (i.e. mapped reads having either the 'reference' or the 'alternate' allele in relation to the called SNP site), with the alternate allele being present in significant minority (Figure 2.1).

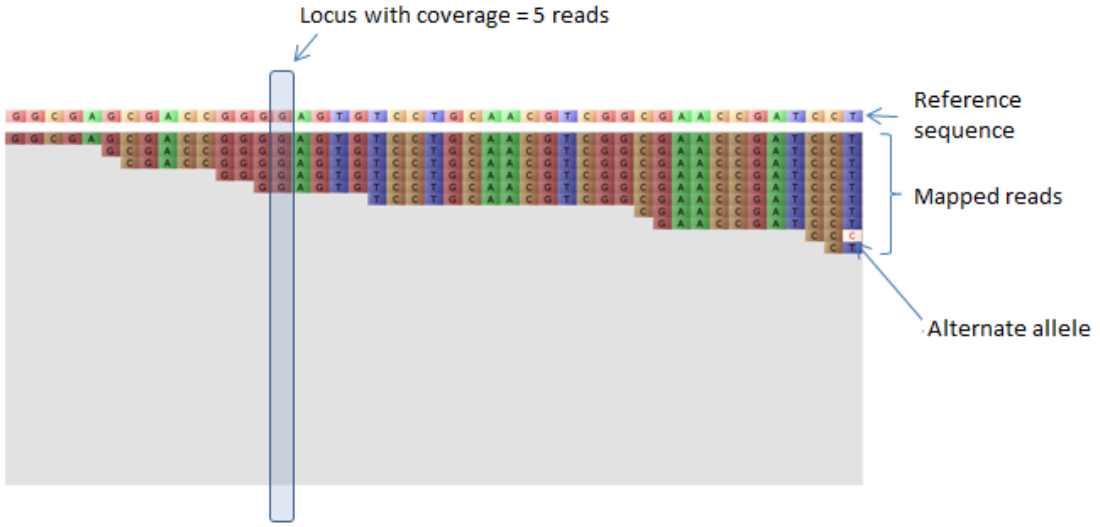

Figure 2.1: Tablet tool (Milne et al., 2010, 2013a) screenshot of an alternate allele 'C' being present in significant minority at a given SNP locus (right side of the figure). The left side of the figure shows an example of a single locus where the coverage — number of reads covering a given position of the reference sequence — is of 5 mapped reads (containing the same allele of the reference). Adapted from A. Golicz (unpublished undergraduate honours project).

This behaviour could be explained, for instance, by the occurrence of sequencing errors in the reads or read cross-mappings due to the existence of very similar regions in the reference. However, the occurrence of homozygous SNPs (i.e. all the reads having the same alternate allele) to the assembled reference sequence was unexpectedly observed (Figure 2.2).

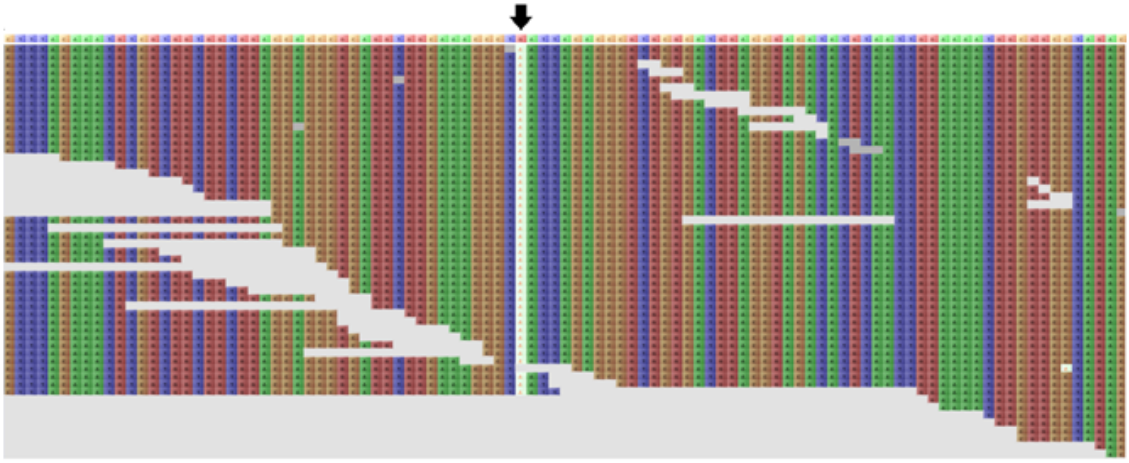

Figure 2.2: Example of a homozygous SNP visualised with the Tablet tool. Adapted from A. Golicz (unpublished undergraduate honours project).

Such occurrences were assumed to be artefactual, in a mapping of reads against a reference sequence produced from the reads themselves, since the reads containing the SNP site should have contributed to the reference. Aiming to uncover this specific mechanism of FP SNP generation, one aspect analysed by Golicz's study focused on the differences between two of the supported mapping modes by the Bowtie tool — "unique" and *all*. The mapping mode relates to the strategy used by the mapping tool for handling reads that could potentially map to more than one location, for example, where closely related members of a gene family are involved (Milne et al., 2013b). As explained in Golicz's study, apart from other factors which also rule the alignment policy, like the number of mismatches allowed and other fine tuning options (Johns Hopkins University, 2009), Bowtie finds all valid read alignments when set to *all* mapping mode. Conversely, under the

"unique" setting, Bowtie supresses all the alignments above a given number of valid alignments specified by the user. In the carried out experiments, for instance, the maximum number of valid alignments for the alignment to be reported was chosen as one. The pattern that emerged from this mapping mode analysis was that many homozygous FP SNPs revealed in "unique" mode often became heterozygous ones in the *all* mode, just like as exemplified in Figure 2.3 below.

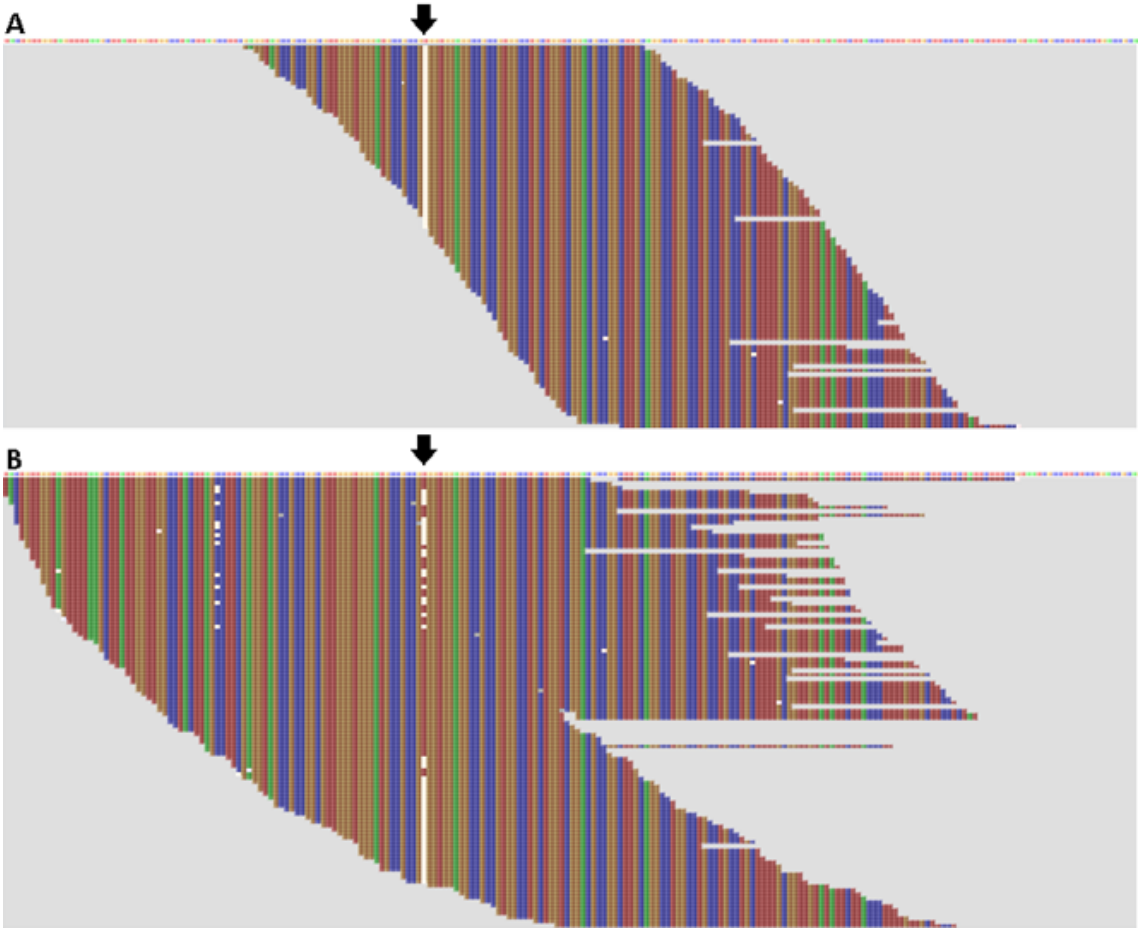

Figure 2.3: (A) Example of a homozygous SNP in the output of the Bowtie "unique" mapping mode visualised with the Tablet tool. (B) The same SNP appears as heterozygous in the Bowtie *all* mapping mode. The SNP locus is indicated by the black arrows. Adapted from A. Golicz (unpublished undergraduate honours project).

The issue was further analysed by an approach which presented an overview of the available reads spanning the region of the SNP during the assembly and mapping stages. The results obtained suggested that the assembler had apparently produced a hybrid sequence from the two types of reads mapped (Figure 2.3(B)). So, apparently, the FP SNP is generated at the mapping stage due to the assembly error. The study then suggested the presence of paralogs — homologous sequences derived by a duplication event from a single sequence (Fitch, 1970, 2000; Jensen, 2001; Kuzniar et al., 2008) — as a possible cause for the FP SNP generation mechanism.

In terms of sequence structure, paralogs may be considered nearly-identical sequences. Phillippy et al. (2008), when proposing the first integrated pipeline for assembly validation — *amosvalidate* — schematised the assembly problem provoked by nearly-identical repetitive sequences (Figure 2.4). According to their explanation, repeats confound the assembly process, which is unable to distinguish reads belonging to distinct copies of the repeat. An additional highlighted issue, specifically related to nearly-identical repeats, is that the assembler cannot differentiate sequencing errors from true polymorphism between repeat copies. This eventually can lead the assembler to place a repetitive read in the wrong copy of a repeat, resulting in the kind of misassembly shown in Figure 2.4. The authors also state that small differences between repeat copies, which are often true SNPs

caused by independently arisen mutations in each copy, sometimes turn out to be useful signatures of misassemblies, as these result in micro-heterogeneities (SNPs) which are correlated across multiple overlapping reads (Figure 2.4(B)) (Phillippy et al., 2008).

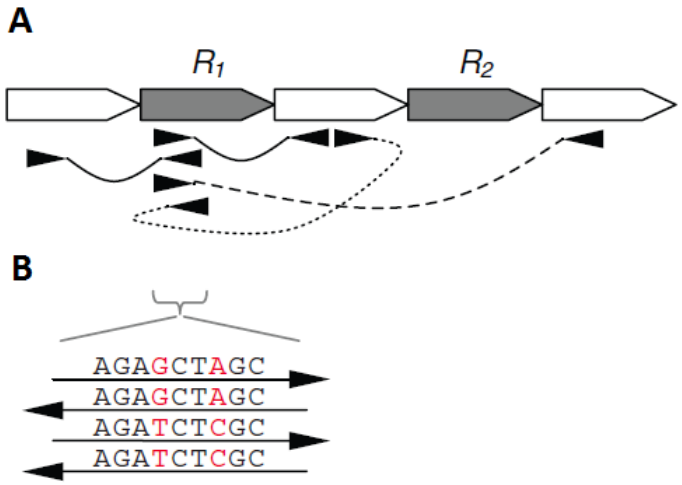

Figure 2.4: Schematic of reads misplaced during assembly, caused by a two copy repeat R, as explained by Phillippy et al. (2008). (A) Occurrences of discordant mate-pair reads. (B) Generation of correlated SNPs. Unique sequence is shown in white while repetitive sequence is shown in gray. Example mate-pairs illustrated as connected arrow heads, where properly oriented ones point towards each other and properly sized pairs are connected by solid lines. Adapted from Figure 1 (Phillippy et al., 2008).

In a similar manner, the putative mechanism for the misassembly of paralogs and consequent FP SNP generation is that *de novo* assembly software produces hybrids of the two very similar sequences, like in the example shown in Figure 2.5 below.

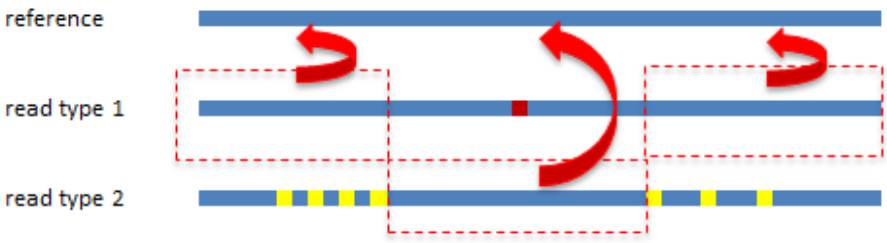

Figure 2.5: The putative mechanism for the reference misassembly origination. During the assembly of the reference sequence, read type 1 contributes its peripheral portions while read type 2 contributes its central part. A hybrid reference sequence is formed from the two similar reads. Adapted from M. Bayer (unpublished material).

In the above case, when mapping is performed, for example, with only a single mismatch allowed, a homozygous SNP *versus* the reference is observed, i.e. all the reads have the alternate allele. When the mapping is carried out with a large number of mismatches allowed, a second group of reads is aligned. These reads feature a significant number of mismatches and are consistent with the existence of a paralogous gene. However, none of the reads match the reference perfectly and the theory is that the base that differs from the reference in read type 1 should in fact not be different from the one in the reference, and, presumably, different on read type 2, suggesting a straight swap of the specific portion in question.

Since, to the best of my knowledge, no automated reporting approach capable of quantifying FP SNPs generated by misassembly mechanism was available (particularly regarding verification whether SNPs originate from paralogs), I designed experiments to answer the following questions:

- In what extent homozygous SNPs, in a mapping of reads against a reference made from the same reads, are caused by misassembly of the reference sequence?

- If/when they are, is the underlying cause for this the existence of paralogs that share one or more very similar stretches of sequence which are capable to confuse the *de novo* assembly tools?

2.2 Testing for paralogs: an experiment with real RNA-Seq data from the barley cultivar Bowman

An automated pipeline was developed to test for FP SNPs and whether they have been caused by paralogs. The approach was to check for the existence of two distinct sets of reads in a mapping to a given transcript assembled, to test the potential true origins of such distinct classes of reads as being paralog-related sequences, and to report the results obtained.

2.2.1 Methods

2.2.1.1 Datasets used

The following datasets were assigned as inputs for the experiment:

- An existing *de novo* transcriptome assembly of barley cultivar Bowman (M. Bayer, unpublished material) — assumed as an effectively homozygous organism. The assembly had been generated with the Trinity transcriptome

assembler (version trinityrnaseq_r2012-06-08; (Grabherr et al., 2011)), from 76 bp single-end Illumina (Illumina, Inc., 2009) whole transcriptome (RNA-Seq) reads, yielding 53,336 transcripts. Some basic assembly statistics, reported by the Trinity utility script TrinityStats.pl (version trinityrnaseq_r20140717), are provided in the Appendix A, subsection A.1.1, item A.1.1.1;

- A subsequent mapping, using the 53,336 assembled transcripts as reference sequences, of the same reads used in the assembly (M. Bayer, unpublished material). Such mapping had been carried out with the Bowtie mapping tool (Langmead et al., 2009), allowing 1 mismatch per read, reporting only uniquely mappable reads, and using the `--best --strata` flag (a more detailed explanation about this flag can be found in the introductory section of Chapter 3);

- A text-file list, with potential homozygous FP SNP sites, which had been generated by a custom SNP calling code originated in Golicz's study;

- A set of 22,651 non-redundant full-length cDNA (FLcDNA) sequences of *Hordeum vulgare* 'Haruna Nijo' two-row malting barley cultivar (Matsumoto et al., 2011), which was used in a further stage of the pipeline execution (for the paralogy test). The Haruna Nijo FLcDNA sequences are considered as a unique resource in the barley community, as they constitute the only reliable source of full-length transcript sequences for barley. They represent

a comprehensive information about the barley gene repertory (Matsumoto et al., 2011). The technology for obtaining them ensures that the product captured is genuinely full-length, their sequencing is Sanger-based, and the subsequent assembly involves the most basic of OLC approach. This means bioinformatics-related artefacts are unlikely and that the resulting sequences are of extremely high quality and value.

2.2.1.2 Software implementation and use

An automated pipeline, employing an approach similar to that used in Golicz's study in terms of inspecting the reads spanning the SNP sites during the assembly and mapping steps, was developed (its workflow is shown further below in Figure 2.6). The rationale behind the pipeline was to have the same original reads mapped back to each SNP location (per transcript) in question, with more relaxed mismatch rate settings, aiming to check for the occurrence of different read classes during assembly time. This, by its turn, would be initially characterised by the presence of the reference and alternate alleles in the group of reads covering the SNP positions under evaluation. Such positions, in the strict mapping, present only the alternate allele for the entire pile of reads. This kind of information would allow to confirm sites, along the reference sequence, for which the *de novo* assembler might have been confounded, consequently resulting in a swap of the base at the locus further detected as a homozygous SNP (as exemplified in Figure

2.5). Additionally, by getting unique representatives of the different classes of reads involved with a given SNP site, such sequences could be (optionally) BLASTed against an annotated database of the organism in question in order to confirm whether they could be related to paralogs or not.

To accomplish this, after proper codification, the pipeline had to implement automated ways of retrieving the original reads used by Trinity software in the original transcriptome assembly, employ relaxed parameter settings of the BWA mapping tool version 0.5.9rc1 (Li and Durbin, 2010) to map all the potential reads participating in the assembly, process intermediate files and calculations using SAMtools version 0.1.18 (Li et al., 2009a) and the Java Picard API (Broad Institute, 2014b), and compute the collected information to properly report the results. The BWA mapper was used to overcome the maximum limit of 3 mismatches allowed hard coded in Bowtie tool, so more relaxed mappings could be obtained. Furthermore, since BWA does not support multi-mapping of reads, these are expected to map to a single location only thus preventing other potential sources of confusion. Values of 5, 10, 20, and 30 mismatches allowed were arbitrarily chosen in order to represent the gradual alleviation of the mismatch rate stringency which could potentially disclose the reads taking part in the assembly.

In more detail, as inputs, the pipeline receives the following from the user:

- The file system path to the original transcriptome assembly file in FASTA format;
- The path to the text file list containing the SNP(s) position(s) per transcript to be explored by the pipeline;
- The path to the directory with the original reads used by Trinity;
- The number of mismatches allowed;
- The number of parallel threads to be used by the BWA mapper;
- The desired name of the output file in which the results should be printed.

In case the user opts to run the paralogy test, this must be informed accordingly along with the path to the target database. The program then parses the transcriptome assembly file and creates a *hash* structure in memory which stores each transcript identifier associated to its respective sequence. A similar structure is built associating each transcript identifier with its corresponding SNP site(s) for exploration within the transcript. By accessing such structures in memory, the program is capable of recreating each needed transcript sequence, having the proper reference and BWA indexes created for the subsequent more relaxed BWA mappings (specified by the user), and iterating over each SNP position per transcript. At each iteration, the original reads used in the assembly are retrieved, a given relaxed mapping is performed, and some basic mapping statistics (e.g.

percentage of covering reads per event) are provided with SAMtools. The reference base information at the position being evaluated is also captured for computing purposes. Subsequently, an auxiliary Java class named *AlleleExtractor*, making usage of resources and associated methods provided by the Java Picard API (e.g. *SAMSequenceRecord*, *SAMRecordIterator*, etc.) is responsible for computing the alleles present in each read overlapping the position under evaluation. Instances of reads with the 'reference' allele are counted as well as the ones of reads containing any 'alternate' allele. After the processing, an output file with the results obtained per iteration is produced. Such file is a tabulated text file which can be easily exported to a spreadsheet software (e.g. Microsoft Excel) for further usage in the downstream analysis.

In a second (optional) stage of the pipeline execution, unique representatives of the reads used in the mappings are compared with the non-redundant FLcDNA sequences (Matsumoto et al., 2011) using the BLASTN tool (Altschul et al., 1990). In the pipeline, valid 'hits' in the FLcDNA sequences 'database' file are considered to be computed, for the characterisation of different classes of reads taking part in the assembly, only when a read matches a given FLcDNA with 100% of similarity and for the entire length of the read. Although being susceptible of producing substantial numbers of false negatives hits, such very stringent approach was chosen in order to assure that a hit is really to the correct FLcDNA and not

to its paralog. This is due to the fact that the FLcDNA sequences come from a different cultivar than the Bowman one, so inherent genetic differences between the two lines are expected. Since no annotation classifying the FLcDNAs into gene families was available, the counting of hits based just on the characterisation of the distinct FLcDNA identifiers was considered as sufficient for this stage of the paralogy test. This was based on the assumptions that each FLcDNA represents a different gene (because of the redundancy removal stage in their preparation) (Matsumoto et al., 2011) and that, in occasions where FLcDNAs share sequence that only differs by a few bases, they are likely to be members of the same gene family/paralogs.

When the user opts to use the tool with the paralogy test, the output reporting file is expanded accordingly to include the information about the BLAST hits to different classes of sequences observed. Figure 2.6 details the workflow of the described pipeline.

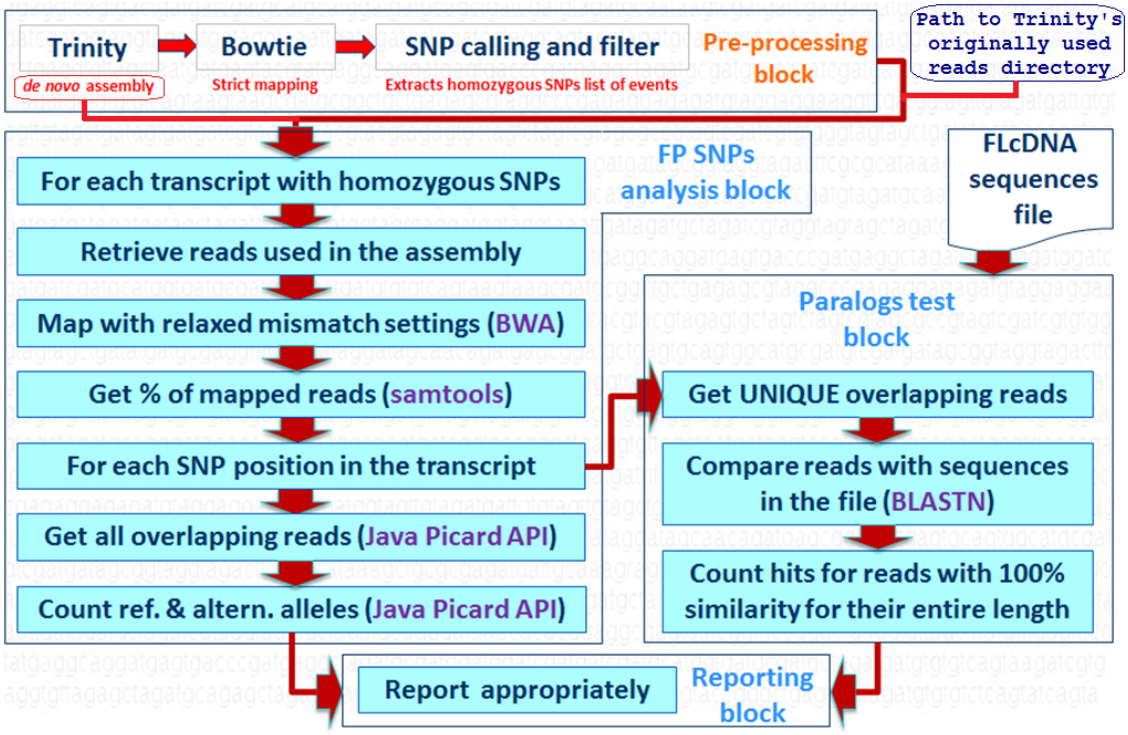

Figure 2.6: The pipeline workflow for the Bowman RNA-Seq dataset experiment. Additional details of the pipeline usage can be seen in the Appendix A, subsection A.1.1, item A.1.1.2. Abbreviations: %: percentage; ref.: reference; altern.: alternate.

2.2.2 Results

The pre-processing stage revealed 473 potential occurrences of FP homozygous SNP sites in the 53,336 *de novo* assembled reference transcripts. Between 386 to 442 of these 473 events (Table 2.1), depending upon the chosen relaxed mapping setting, were reported by the pipeline as having more than one group of reads overlapping the SNP site. In these cases, one of the groups consisted of reads containing the same allele as the reference, while the remainder were comprised by reads containing an alternate allele (Figure 2.7).

Table 2.1: Summary of results from the four different scenarios of mismatches allowed (n = 5, 10, 20, and 30) for the BWA mappings (for the barley dataset) after the run of the pipeline tool with the BLAST search feature turned ON.

Pipeline summary	$n = 5$	$n = 10$	$n = 20$	$n = 30$
# cov. reads (in avg.) when ref. and alt. alleles WERE present in the reads set	∼187	∼198	∼221	∼279
# cov. reads (in avg.) when ref. allele NOT present in the reads set	∼16	∼21	∼11	∼11
# cases in which ref. and alt. alleles WERE present in the reads set	386	414	435	442
# cases in which the ref. allele WAS NOT present in the reads set	87	59	38	31
Breakdown of cases with ref. and alt. alleles present				
# cases with ONLY the ref. allele and ONE type of alt. allele	352	347	259	179
# cases with three or more alleles present	34	67	176	263
FLcDNA database test output				
# cases with ref. and alt. alleles present in the reads set and hitting ≠ FLcDNAs	113	129	140	154

Abbreviations: n: BWA mismatch allowed rate; #: Number of; cov.: covering; avg.: average; ref.: reference; alt.: alternate; ∼: approximately; ≠: different.

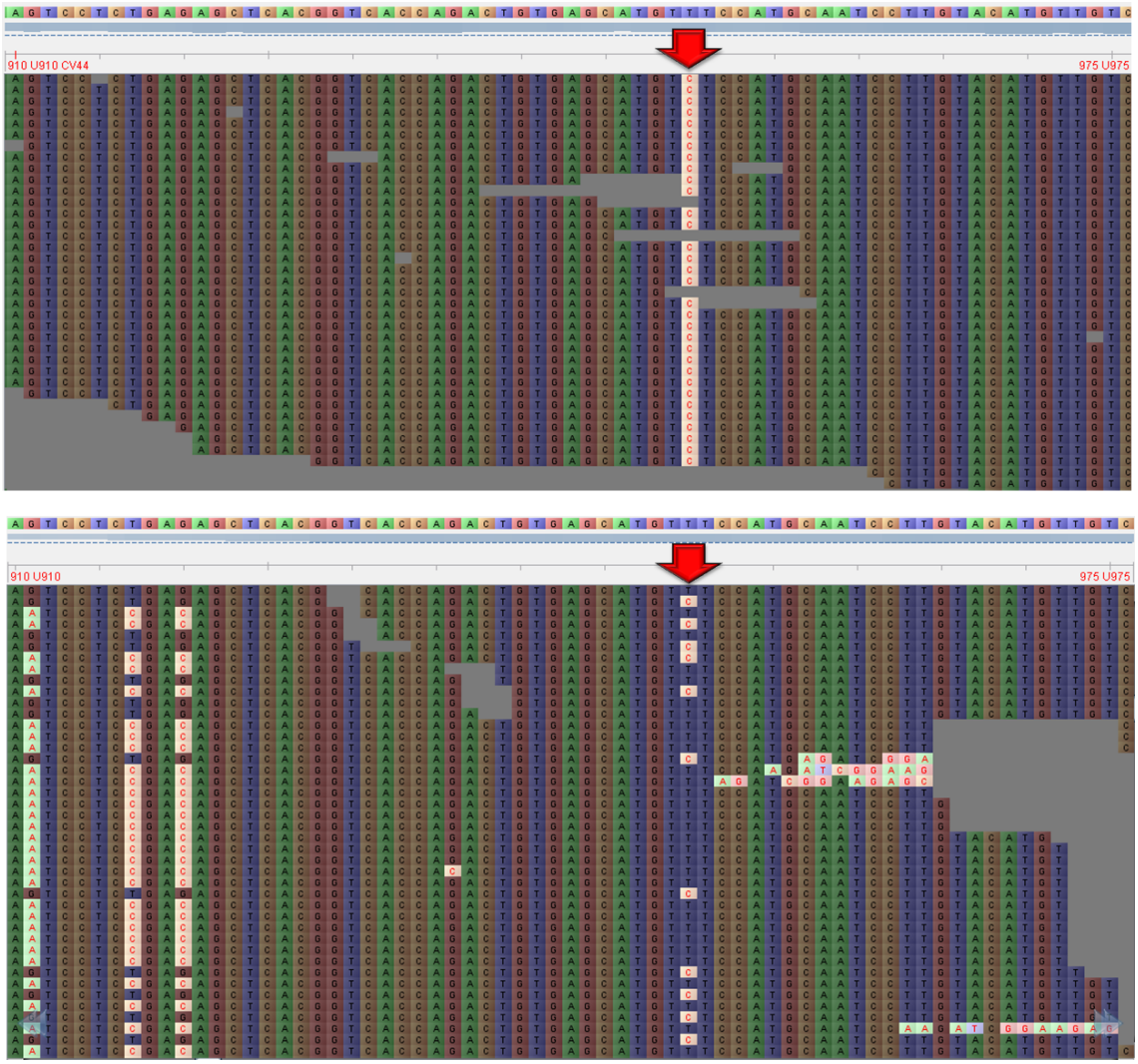

Figure 2.7: Screenshots taken with the Tablet graphical viewer showing the mappings obtained for the reference transcript comp13964_c0_seq2 at position 950 (highlighted by the red arrows). At the top, the strict mapping shows the homozygous SNP site with only the alternate allele 'C'. Bottom: the different group of reads revealed by the relaxed mapping which were potentially used by the *de novo* assembler software. One group of reads has the alternate allele 'C' and the remainder have the reference allele 'T'.

In order to check for the existence of paralogs which could explain the different classes of reads found during the assembly of the reference sequence, each set of reads taking part in the mappings for FP SNP sites was BLASTed against the

22,651 sequences corresponding to non-redundant Haruna Nijo barley FLcDNAs (Matsumoto et al., 2011). The results (Table 2.1) confirmed that a range of 29 to 35% of events had different classes of reads (in the subset used in the transcript assembly) scoring 'perfect' matches — the entire length of the query read had 100% similarity to the subject — with distinct FLcDNA types. Such different types of reads, containing either the reference or the alternate allele, were gradually revealed in response to the more relaxed BWA mapping settings applied by the pipeline.

It is possible that the above results could be somewhat restrained due to the genetic differences between the cultivars used (Bowman *versus* Haruna Nijo) and/or to any eventual bias associated to the fact of different classes of reads under evaluation being more prone to hit particular subsets of FLcDNAs. Therefore, two additional experiments were performed to verify the trend of FLcDNA hits observed in response to different sizes of the FLcDNAs database and

considering only the data from the most stringent run of the pipeline (5 mismatches allowed in BWA):

(i) Nine files were created to represent the contents of the original database file by simulating a random subset of the FLcDNA sequences. For example, a file was created with 10% of the sequences present in the original file, chosen at random, another file was created with 20% of the sequences, a third file with 30% of the

sequences, and so on (with the last file comprising 90% of the sequences). In this scenario, named here as "RANDOM" experiment, a given FLcDNA sequence could eventually be present in one file, or in more than one file, or in none of the files at all, due to the random nature of the selection process;

(ii) In the second experiment, named here as "NON-RANDOM" experiment, the idea was similar, but each file contained an increasing proportion of sequences picked from the original file in a deterministic manner. To improve the resolution of this particular experiment, files comprising 25 and 75% of the sequences were also made available. In this experiment, sequences present in the first file would also be present in the second, which, in turn, would be present in the third one, and so on. Table 2.2 shows the results registered for the increasing dataset of FLcDNAs created in random fashion while Table 2.3 shows the equivalent results for the non-random sampling.

Table 2.2: Comparison of nine different runs of the pipeline using the same mismatch setting, but varying numbers of the randomly chosen FLcDNAs in the BLAST target database.

	% of availability of FLcDNAs								
Pipeline summary	10	20	30	40	50	60	70	80	90
# cov. reads (in avg.) when ref. and alt. alleles WERE present	∼187								
# cases in which ref. and alt. alleles WERE present	386								
FLcDNA database test output									
# cases with ref. and alt. alleles present and hitting ≠ FLcDNAs	1	7	7	31	47	70	48	97	103

Abbreviations: %: percentage; #: Number of; cov.: covering; avg.: average; ref.: reference; alt.: alternate; ∼: approximately; ≠: different.

Table 2.3: Comparison of twelve different runs of the pipeline using the same mismatch setting, but varying numbers of chosen FLcDNAs in the BLAST target database in a deterministic manner.

	% of availability of FLcDNAs											
Pipeline summary	10	20	25	30	40	50	60	70	75	80	90	100
# cov. reads (in avg.) when ref. and alt. alleles WERE present	∼187											
# cases in which ref. and alt. alleles WERE present	386											
FLcDNA database test output												
# cases with ref. and alt. alleles present and hitting ≠ FLcDNAs	3	14	14	24	38	62	66	69	79	84	92	113

Abbreviations: %: percentage; #: Number of; cov.: covering; avg.: average; ref.: reference; alt.: alternate; ∼: approximately; ≠: different.

2.3 Testing the misassembly with a *de novo* genome assembler

To further expand the investigation of misassembly as a cause of FP SNPs, another experiment was designed to check for the occurrence of this phenomenon for a *de novo* assembled genome. A different line of thought was taken with regards to the evaluation of the reads taking part in the assembly process. Some *de novo* genome assemblers, e.g. MIRA (Chevreux et al., 1999), the Roche 454 Newbler assembler (Margulies et al., 2005), and Velvet (Zerbino and Birney, 2008), are capable of providing information related to the assembly process in the form of ACE or AFG files (i.e. read and consensus/contig information). Additionally, some NGS read simulators (e.g. Sherman (Babraham Bioinformatics, 2013b) and SimSeq (St. John, 2014)) provide read origin information as part of the read name, and this was also taken into account in the experiment.

2.3.1 Methods

2.3.1.1 Datasets used

The five chromosome sequences of *Arabidopsis thaliana*, available at
The Arabidopsis Information Resource (2011a) (via NIH (2001)), were used as the template sequences to generate the simulated reads for the experiment. The Sherman read simulator version 0.1.6 (Babraham Bioinformatics, 2013b) was used to generate haploid, error-free, platform-independent paired-end reads from each

of the mentioned chromosome sequences (see Appendix A, subsection A.1.2, item A.1.2.1). Reads of 150 bp in length were produced with 100-fold coverage depth. The five read datasets obtained from the respective five *A. thaliana* chromosomes were combined to form a single paired-end read dataset.

The 150 bp read length value was chosen as a reasonable representative of current NGS scenarios, since it features in a large number of ongoing sequencing projects which use Illumina HiSeq reads as their primary source of sequence. The current maximum read length for this is 150 bp (Illumina, Inc., 2014a). Also, even projects involving the assembly of very large, complex genomes such as wheat (IWGSC, 2014) use reads as short as this or even shorter (barley (IBGSC, 2012), norway spruce (Nystedt et al., 2013)) as their primary source of sequence.

A reference sequence for the read mapping was then assembled *de novo* from the 150 bp read dataset, using the Velvet assembler, version 1.2.10 (Zerbino and Birney, 2008) (see Appendix A, subsection A.1.2, item A.1.2.2, for detailed commands). Velvet was chosen for this specific experiment due to its capability of generating the AFG text-based file mentioned above. The quality assessment tool for genome assemblies, QUAST version 2.1 (Gurevich et al., 2013), was employed to evaluate the assembly. Its report is detailed in Appendix A, subsection A.1.2, item A.1.2.3.

After the production of the reference assembly, the read dataset was mapped against it with Bowtie2 version 2.2.1 (Langmead and Salzberg, 2012), applying

a stringent mismatch rate setting of 3 mismatches (in 150 bp read length; i.e. 1 mismatch per 50 bp) (see Appendix A, subsection A.1.2, items A.1.2.4 and A.1.2.5, for respective detailed mapping commands and results).

Variant calling was then performed with FreeBayes version 0.9.9 (Garrison and Marth, 2012) (see Appendix A, subsection A.1.2, item A.1.2.6, for the used command and parameters). The resulting VCF file from FreeBayes was processed by existing tallying/classifier Java code (M. Bayer, unpublished material) which counted and classified the called SNPs in three main categories: "multi-allelic", "homozygous", and "heterozygous". The outcome was reported in a text file that was subsequently filtered to retrieve only the homozygous SNP occurrences. This final list was used as the input for a refactored version of the pipeline described in Section 2.2; this time conceived to take advantage of the AFG file information provided by the Velvet *de novo* assembler (see Appendix A, subsection A.1.2, item A.1.2.7, for the used command and parameters). This allowed the pipeline to retrieve the original reads used in the assembly and to check for potential different classes of reads involved. Relaxed mappings using such reads were also carried out as means of comparison (see Appendix A, subsection A.1.2, items A.1.2.8 and A.1.2.9, for respective detailed mapping commands and results).

In order to test for paralogs and to identify genomic features corresponding to the reads involved in the assembly of contigs with homozygous SNPs, the

pipeline was fed a BLAST database composed of coding sequences (CDS) and intergenic regions retrieved from the *A. thaliana* annotation. The gene model (The Arabidopsis Information Resource, 2011b) and intergenic regions files (The Arabidopsis Information Resource, 2010) were combined into a single FASTA file of coding and intergenic sequences. This file was then used to build the BLAST database with the makeblastdb command.

2.3.1.2 Software implementation and use

The pipeline described in Section 2.2 was refactored to make use of the AFG text file provided by Velvet. In its output directory, depending on what is specified on the command line of its intermediate steps *velveth* and *velvetg*, Velvet produces a number of files, such as *Sequences* and *velvet_asm.afg* (Zerbino, 2010). The former is a modified FASTA file which contains the original read names and the corresponding read ID numbers assigned by Velvet. The latter contains all the assembly scaffolding information explicitly, specifically the information on the inferred mapping of the reads onto the contigs. Thus, the pipeline was modified to parse the AFG file, get the read ID numbers related to a given assembled contig, and map these identifiers in the *Sequences* file, retrieving the targeted reads. To improve the speed of the overall process and avoid input/output overload, the pipeline converts the Velvet *Sequences* file onto a hash table data structure in the beginning of each run. Once created, the data structure is easily accessed

by the pipeline at each contig assembly "search for reads" iteration. Also, to improve the management of the typically huge AFG file, the Velvet accessory script "asmbly_splitter.pl" (developed by Simon Gladman and cited in EMBL-EBI (2008)) was incorporated into the pipeline in order to produce a separate AFG file for every contig analysed.

Finally, for the sake of consistency, the pipeline was adapted to use Bowtie2 in the relaxed mappings since the original strict mapping for this new experiment was done with that tool. Furthermore, Bowtie2 is also a widely used alignment tool (Farrer et al., 2013) which allows great parametrisation flexibility. Figure 2.8 shows the workflow of the refactored pipeline for the *A. thaliana* dataset experiment.

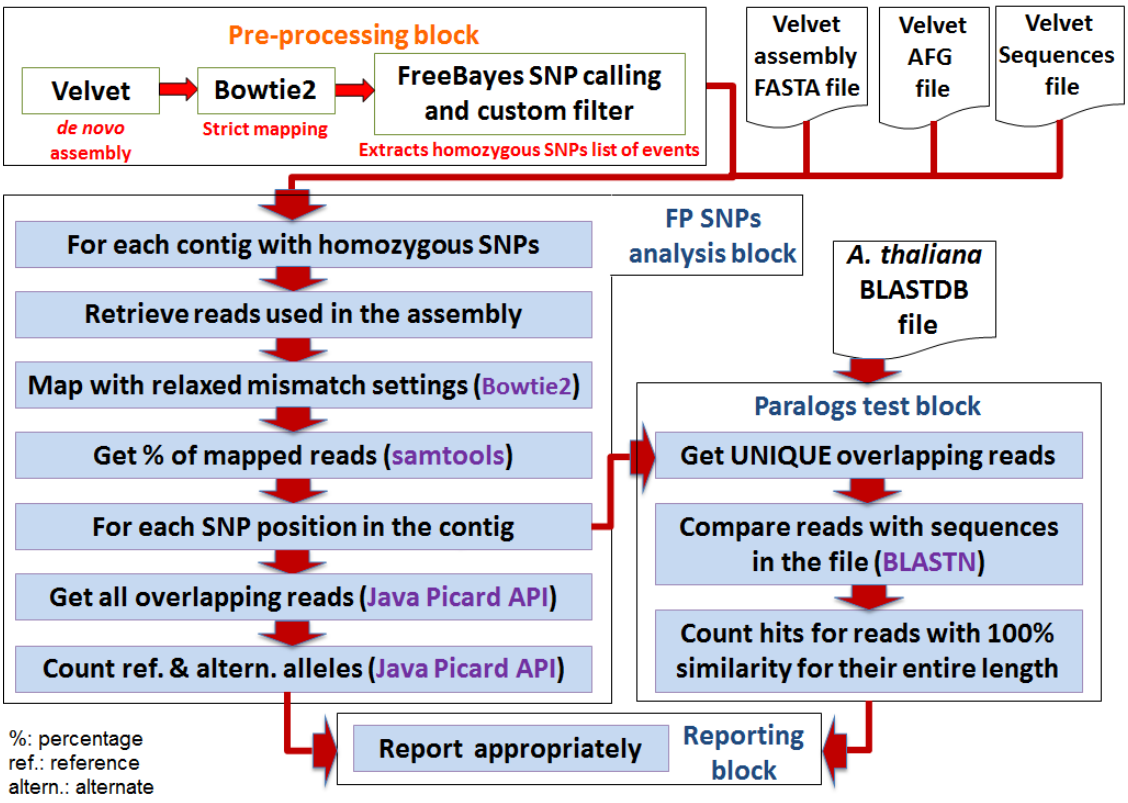

Figure 2.8: The refactored pipeline workflow for the *A. thaliana* dataset experiment.

2.3.2 Results

The pre-processing stage revealed 72 potential occurrences of FP homozygous SNP sites within 7,042 *de novo* assembled reference contigs. Out of these 72 events, a range from 32 to 34 (Table 2.4), depending upon the mismatch rate, were reported by the pipeline as having different groups of reads overlapping the SNP site. As shown previously (Section 2.1), these cases were characterised by one group of reads containing the same allele as the reference, while the other reads contained the alternate allele (Figure 2.9).

Table 2.4: Summary of results from the four different scenarios of mismatches allowed (n = 5, 10, 20, and 30) for the Bowtie2 mappings (for the *A. thaliana* dataset) after the run of the pipeline tool with the BLAST search feature turned ON.

Pipeline summary	$n = 5$	$n = 10$	$n = 20$	$n = 30$
# cov. reads (in avg.) when ref. and alt. alleles WERE present in the reads set	∼54	∼76	∼109	∼134
# cov. reads (in avg.) when ref. allele NOT present in the reads set	∼84	∼85	∼88	∼90
# cases in which ref. and alt. alleles WERE present in the reads set	32	32	33	34
# cases in which the ref. allele WAS NOT present in the reads set	40	40	39	38
Breakdown of cases with ref. and alt. alleles present				
# cases with ONLY the ref. allele and ONE type of alt. allele	31	31	31	32
# cases with three or more alleles present	1	1	2	2
A. thaliana annotation database test output				
# cases with ref. and alt. alleles present and hitting ≠ regions of the database	31	31	32	33

Abbreviations: n: BWA mismatch allowed rate; #: Number of; cov.: covering; avg.: average; ref.: reference; alt.: alternate; ∼: approximately; ≠: different.

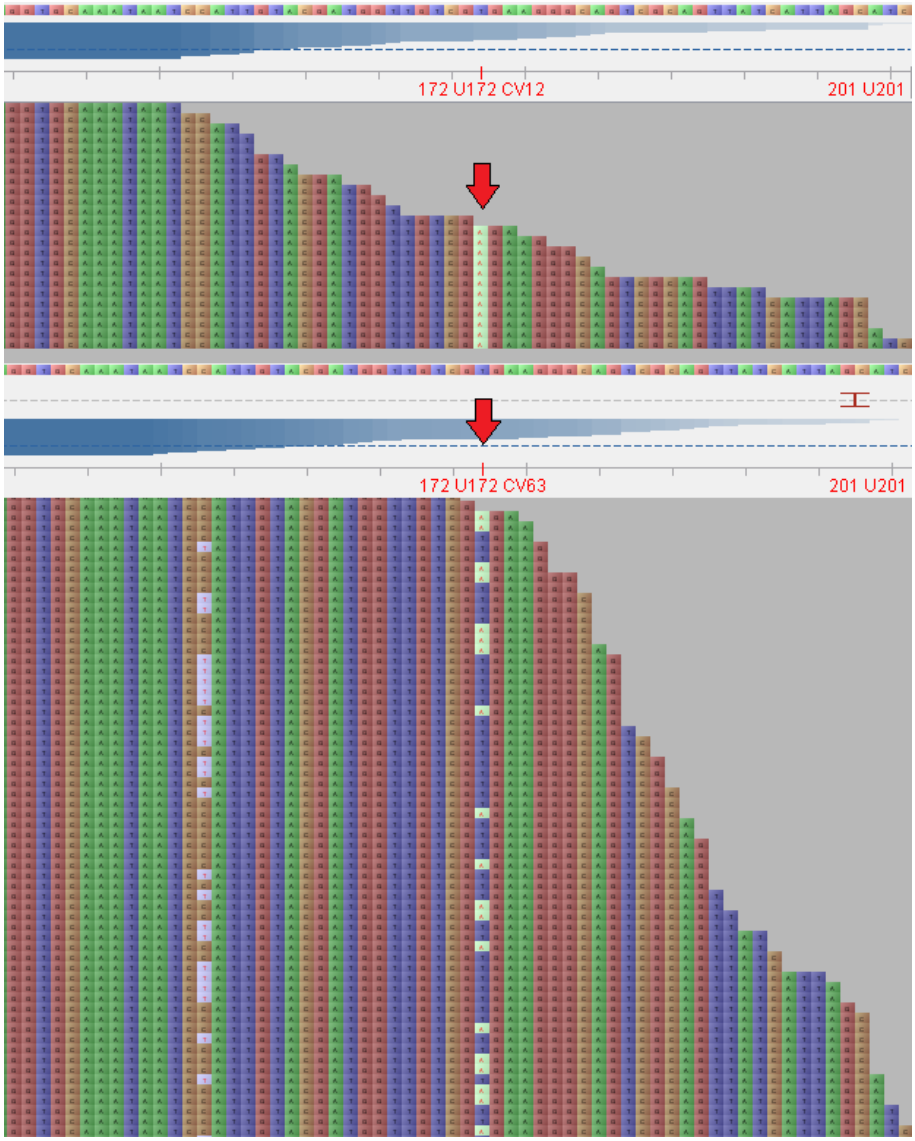

Figure 2.9: Screenshots taken with the Tablet graphical viewer showing the mappings obtained for the reference contig NODE_6286 at position 172 (highlighted by the red arrows). At the top, the strict mapping (1 mismatch allowed per 50 bp read length) shows the homozygous SNP site with only the alternate allele 'A'. Bottom: the different group of reads revealed by the relaxed mapping (5 mismatches allowed per 50 bp read length) which were potentially used by the *de novo* assembler software. One group of reads has the alternate allele 'A' and the remainder reads have the reference allele 'T'.

In the scenario shown in the figure, the reads containing the alternate allele

all came from chromosome 4, as indicated by the read name information. Reads containing the reference allele come from chromosomes 1 and 3, as shown in more detail in Figure 2.10.

Figure 2.10: Screenshot showing the labels of reads revealed by the relaxed mapping (with 5 mismatches allowed per 50 bp read length) for contig NODE_6286 at position 172 (highlighted by the red arrow). The outlined read at the top has the alternate allele 'A' and originates from chromosome 4. The outlined read in the centre contains the reference allele 'T' and comes from chromosome 1. This class of reads also has another variant ('T' allele) at position 153 (highlighted by the yellow arrow) and is completely out of phase with the class of reads with the 'A' variant from chromosome 4. The read outlined at the bottom of the figure belongs to chromosome 3 and has both the reference alleles 'T' at position 172 and 'C' at position 153. This third class of reads is also out of phase with the other classes.

The pattern is also confirmed when directly visualising the AFG file portion related to the contig (Figure 2.11).

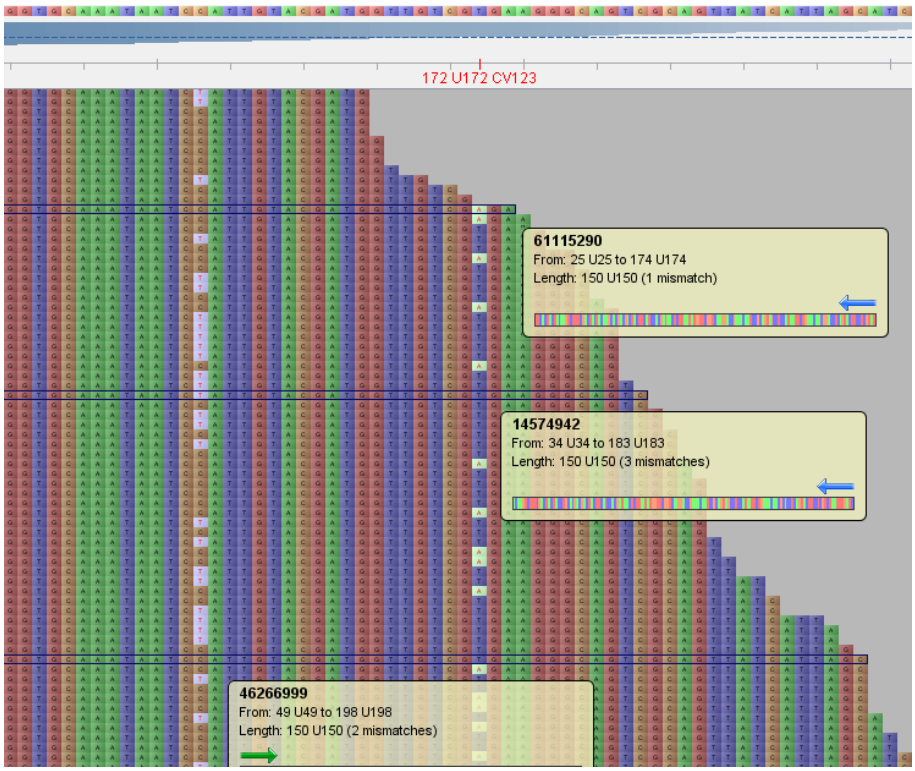

Figure 2.11: Screenshot showing the AFG file portion related to contig NODE_6286 with three distinct classes of reads taking part in the assembly (outlined in dark blue). When mapping the Velvet read IDs from the AFG file to the Velvet *Sequences* file, read 61115290 is the original 6029049_Chr4:1839004-1839357_R2, read 14574942 is 7287471_Chr1:13132292-13132510_R2, and read 46266999 is 6424847_Chr3:12231536-12231778_R1, just as observed in the relaxed mapping previously shown.

In the figure, three distinct classes of reads which took part in the assembly are highlighted. For ease of comparison, Figure 2.12 summarises the scenarios observed in the strict mapping, the relaxed mapping, and the AFG file, in terms of reads covering the SNP site location for the specific contig region.

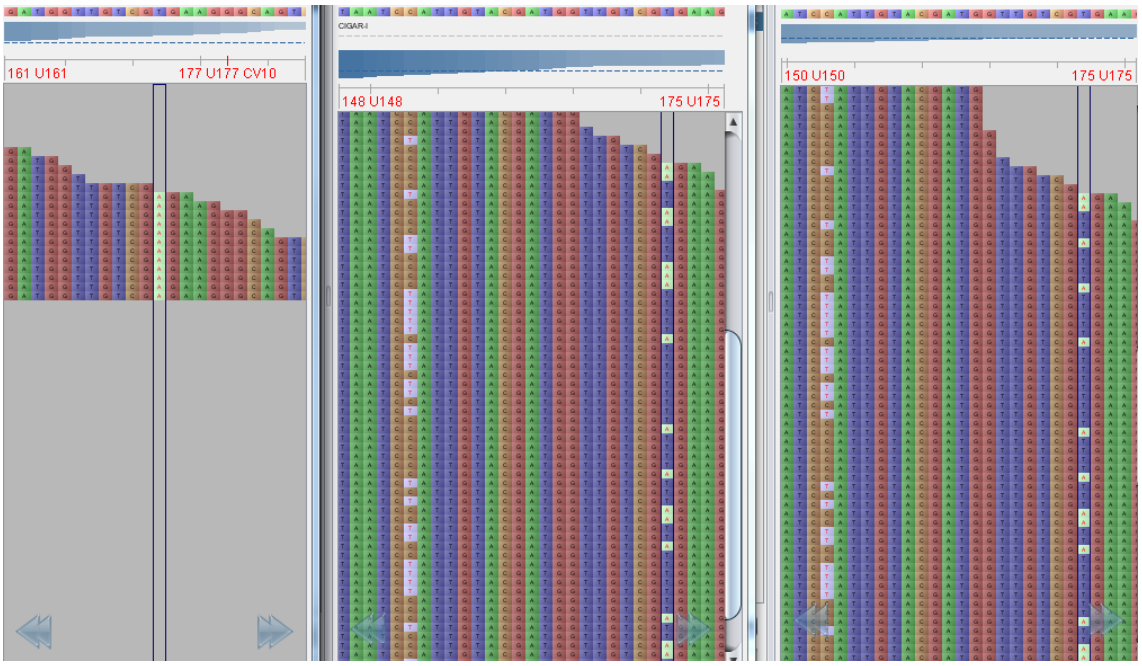

Figure 2.12: Screenshots showing three different observed scenarios for contig NODE_6286, placed side by side for comparison purposes. On the left, the strict mapping (1 mismatch allowed per 50 bp read length) has a homozygous SNP at position 172. In the middle part of the figure, the relaxed mapping (5 mismatches allowed per 50 bp read length) produced by the analysis pipeline uncovers a heterozygous pattern due to the potential reads used in the *de novo* assembly. On the right, the corresponding AFG file visualisation confirms the different classes of reads being used in the assembly of the contig for that specific location.

Figures 2.13 and 2.14 illustrate more examples of the reference misassembly manifesting itself in other contigs.

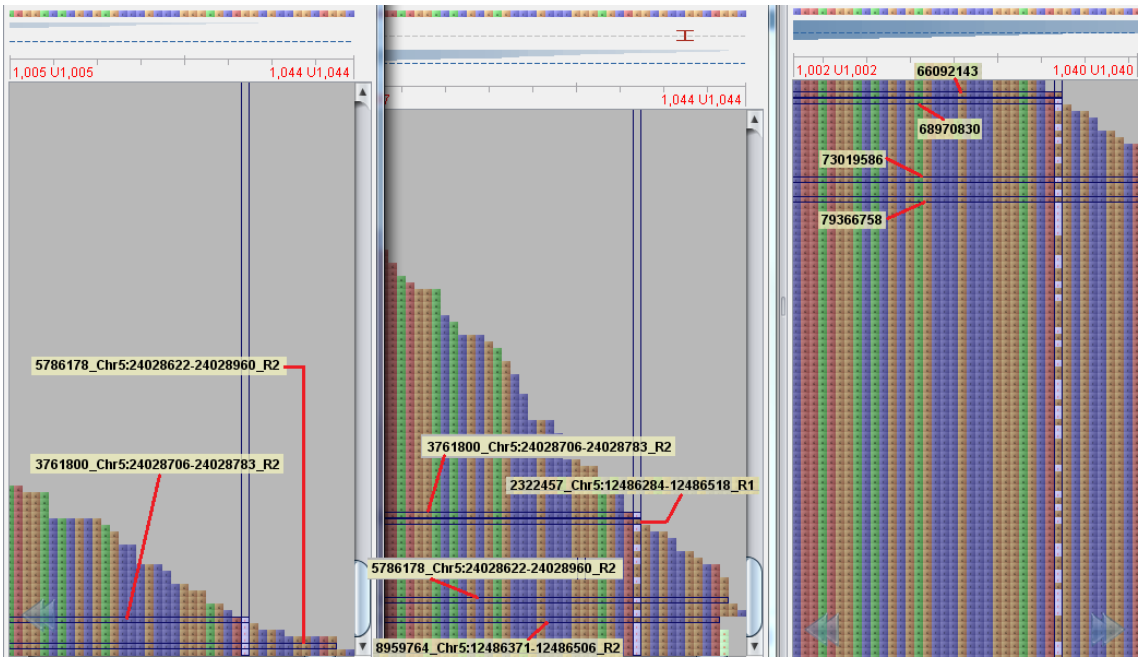

Figure 2.13: Screenshots showing three different scenarios for contig NODE_18482, placed side by side for comparison purposes. On the left, the strict mapping presents a homozygous SNP at position 1,032 (vertical column highlighted in dark blue). Reads 3761800_Chr5:24028706-24028783_R2 and 5786178_Chr5:24028622-24028960_R2 are highlighted as examples of the ones containing the alternate allele 'T'. In the middle part of the figure, the relaxed mapping uncovers a heterozygous pattern due to the potential reads used in the *de novo* assembly. Complementing the reads mentioned before, reads 2322457_Chr5:12486284-12486518_R1 and 8959764_Chr5:12486371-12486506_R2 are highlighted as examples of the ones containing the reference allele 'C'. Although originally from the same chromosome, they come from a different location around 12 Megabases apart. On the right, the corresponding AFG file visualisation confirms the different classes of reads being used in the assembly of the contig for that specific location. From top to bottom, based on Velvet's *Sequences* file, highlighted read IDs are correspondent to the following reads shown in the middle part of the figure: 66092143 - 2322457_Chr5:12486284-12486518_R1; 68970830 - 3761800_Chr5:24028706-24028783_R2; 73019586 - 5786178_Chr5:24028622-24028960_R2; 79366758 - 8959764_Chr5:12486371-12486506_R2.

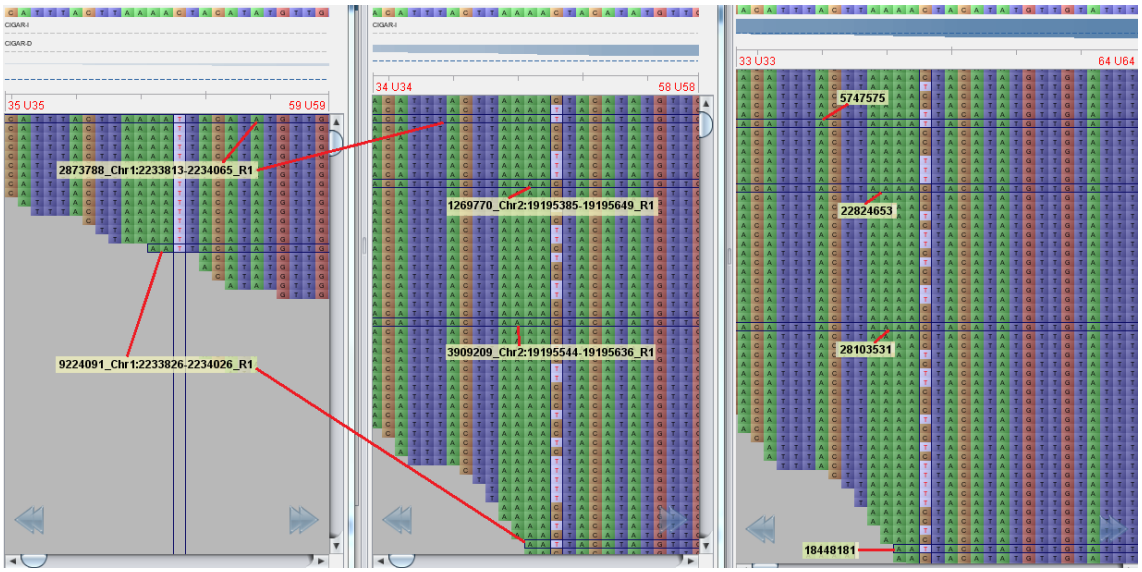

Figure 2.14: Screenshots showing three different scenarios for contig NODE_3278, placed side by side for comparison purposes. On the left, the strict mapping presents a homozygous SNP at position 48 (vertical column highlighted in dark blue). Reads 2873788_Chr1:2233813-2234065_R1 and 9224091_Chr1:2233826-2234026_R1 are highlighted as examples of the ones containing the alternate allele 'T'. In the middle part of the figure, the relaxed mapping uncovers a heterozygous pattern due to the potential reads used in the *de novo* assembly. Complementing the reads mentioned before, reads 1269770_Chr2:19195385-19195649_R1 and 3909209_Chr2:19195544-19195636_R1 are highlighted as examples of the ones containing the reference allele 'C'. These latter come from chromosome 2 while the former come from chromosome 1. On the right, the corresponding AFG file visualisation confirms the different classes of reads being used in the assembly of the contig for that specific location. From top to bottom, based on Velvet's *Sequences* file, highlighted read IDs correspondence is as follows: 5747575 - 2873788_Chr1:2233813-2234065_R1; 22824653 - 1269770_Chr2:19195385-19195649_R1; 28103531 - 3909209_Chr2:19195544-19195636_R1; 18448181 - 9224091_Chr1:2233826-2234026_R1.

As shown in Table 2.4, 2 out of the 72 homozygous SNPs were gradually revealed by the pipeline as containing either the reference or alternate alleles in the reads traversing the SNP positions. One of these cases (NODE_13084) was only uncovered by a more relaxed mapping of 30 mismatches allowed. In

the other one (NODE_13209), different classes of reads were only revealed by the relaxed mappings of 20 and 30 mismatches allowed. For 38 cases (related to the most relaxed mapping scenario), the analysis pipeline did not report the presence of any potential different class of read (with also the reference allele) taking part in the assembly. In other words, for these events, only the alternate allele was present in the read alignments, regardless of how relaxed the mapping was. Nevertheless, when checking the corresponding AFG file for each of these 38 cases, it was observed that 3 had, in fact, different classes of reads involved in the assembly time (NODE_9214, NODE_523, and NODE_10166), but this was not detected by the pipeline at any mismatch setting. Examples of such cases, plus the 2 gradually uncovered, are shown in composite Figure 2.15.

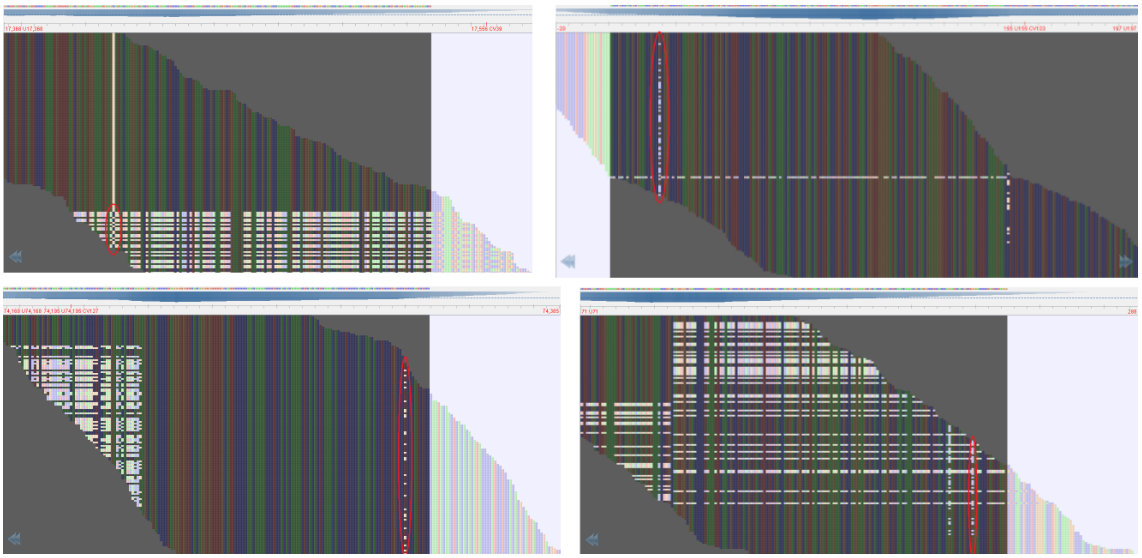

Figure 2.15: Screenshots showing the cases undetected or only partially detected by the pipeline, even though containing different classes of reads involved in the *de novo* assembly. From the upper left corner of the figure, in clockwise direction, the corresponding AFG files for contigs NODE_523, NODE_9214, NODE_13209, and NODE_13084 with the SNP site correspondent position highlighted by red elipses. NODE_10166 (not shown here for simplicity) had a similar pattern of the one presented by NODE_9214.

For the remaining 35 cases out of the 38 mentioned above, homozygous patterns were consistently observed, in the strict and any of the relaxed mappings as well as in the AFG files, exemplified by Figures 2.16 and 2.17.

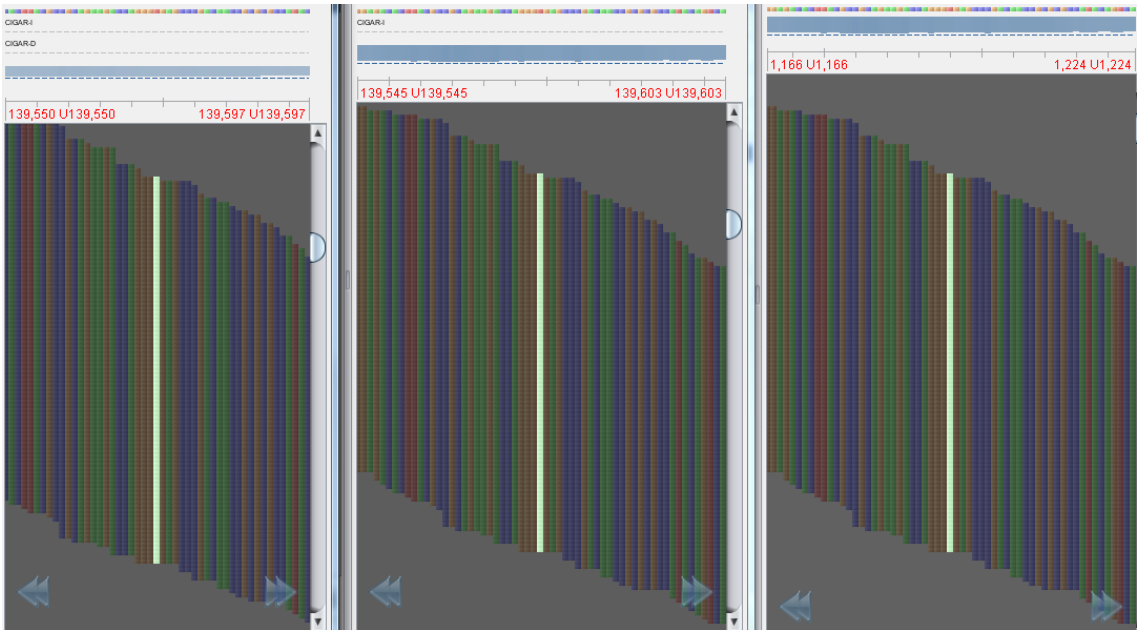

Figure 2.16: Screenshots showing three different scenarios observed for contig NODE_19266, placed side by side for comparison purposes. On the left, the strict mapping (1 mismatch allowed per 50 bp read length) presents the homozygous SNP at position 139,574. In the middle part of the figure, the same homozygous pattern is observed in the relaxed mapping (5 mismatches allowed per 50 bp read length shown) produced by the analysis pipeline. On the right, the homozygous pattern still remains visualised in the corresponding AFG file. When mapping the Velvet read IDs from the AFG file to the Velvet *Sequences* file, all of them were confirmed to have originated in the same region of chromosome 5.

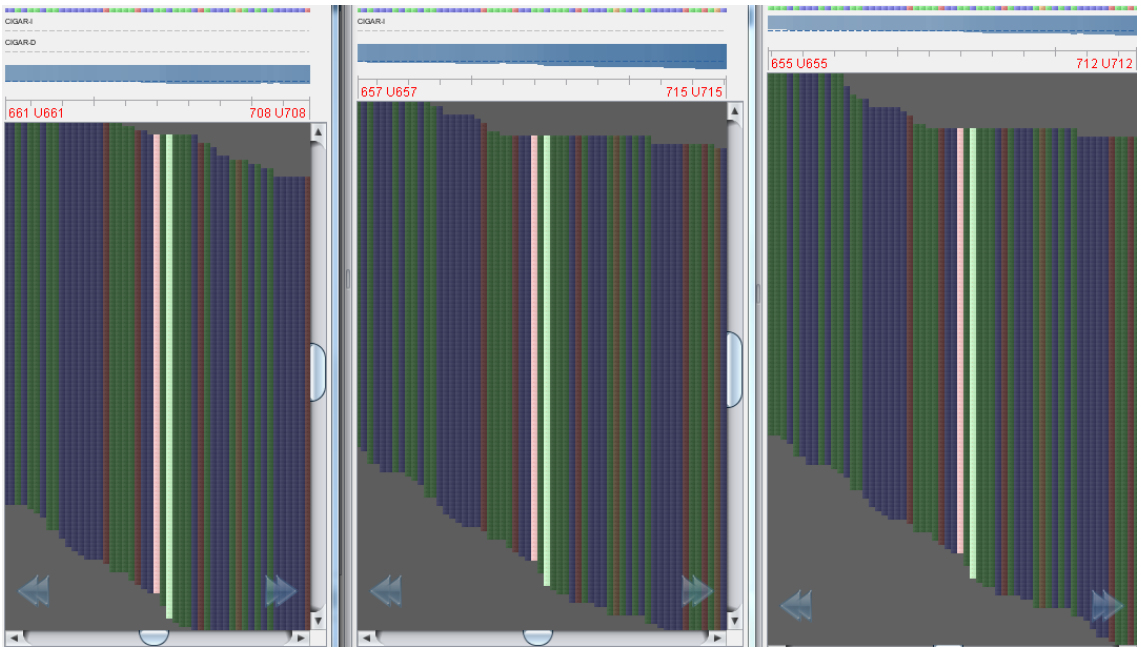

Figure 2.17: Screenshots showing three different scenarios observed for contig NODE_8802, placed side by side for comparison purposes. On the left, the strict mapping (1 mismatch allowed per 50 bp read length) presents homozygous SNPs at positions 685 and 687. In the middle part of the figure, the same homozygous patterns are observed in the relaxed mapping (5 mismatches allowed per 50 bp read length shown) produced by the analysis pipeline. On the right, the homozygous patterns still remain visualised in the corresponding AFG file. When mapping the Velvet read IDs from the AFG file to the Velvet *Sequences* file, all of them were confirmed to have originated in the same region of chromosome 3.

For the 32-34 cases of either reference or alternate alleles identified (Table 2.4), the paralog check resulted in 33 instances of 'perfect' matches for each of the different classes of reads BLASTed against the *A. thaliana* annotation database file. Table 2.5 summarises the results for 5 of these and the paralogy test results for the complete set of 34 cases with either the reference or the alternate alleles present in the reads sets, considering the most relaxed mapping scenario, are shown in Appendix A, subsection A.1.2, item A.1.2.10.

Table 2.5: Five cases of *A. thaliana* annotation BLASTDB hits found by the pipeline for the cases with either the reference or the alternate alleles present in the reads sets considering the most relaxed mapping scenario.

contig	length	SNP pos.	ref. allele	ref. allele hit(s)	alt. allele(s) hit(s) / (allele)
NODE_18482	1,044	1,032	C	AT5G33234.1 - t.e.g. - chr5:12483583-12487437	AT5G59640.1 - t.e.g. - chr5:24025992-24030772 / (T)
NODE_1136	344	304	G	AT5G36870.1 - glucan synthase-like 9 - chr5:14518316-14533930	AT2G15310-AT2G15318 - int. - chr2:6655502-6664759 / (A)
NODE_3278	308,548	48	C	AT2G46710-AT2G46720 - int. - chr2:19194850-19197382	AT1G07270-AT1G07280 - int. - chr1:2232898-2238086 / (T)
NODE_2297	1,123	1,094	T	AT5G28526.1 - t.e.g. - chr5:10515532-10521998	AT4G08060.1 - t.e.g. - chr4:4924776-4928259 / (A)
NODE_2026	388	385	A	AT1G43110.1 - pseudogene, putative polygalacturonase (*Phleum pratense*) - chr1:16223981-16225810	AT1G43120.1 - pseudogene, putative polygalacturonase protein allergen (*Cynodon dactylon*) - chr1:16227318-16227779 / (T)

Abbreviations: pos.: position; ref.: reference; alt.: alternate; t.e.g.: transposable element gene; int.: intergenic region; chr: chromosome.

Based on the Appendix A's Table A.5, taking into consideration the BLAST database annotation as well as the read labels provided by the read simulator (which contain the read origin information regarding chromosome and region range from where the read was sampled), the following summary table (Table 2.6) can be extracted in terms of how the different classes of reads hit different regions or even different chromosomes in the experiment.

Table 2.6: Summary of BLAST database hit counts for the *A. thaliana* paralogy test in the most relaxed mapping scenario.

BLAST database hit characteristic	Number of occurrences
t.e.g *vs* t.e.g in $\neq$ chrm.	9
t.e.g *vs* t.e.g in $\neq$ region of the same chrm.	3
t.e.g *vs* t.e.g in $\neq$ region of the same chrm. OR in $\neq$ chrm.	3
int. *vs* int. in $\neq$ chrm.	7
int. *vs* int. in $\neq$ region of the same chrm.	2
int. *vs* int. in $\neq$ region of the same chrm. OR in $\neq$ chrm.	5
protein *vs* int. in $\neq$ chrm.	1
pseudogene *vs* pseudogene in $\neq$ region of the same chrm.	1
miscellaneous cases	2
no reference allele hit	1

Abbreviations: t.e.g.: transposable element gene; *vs*: *versus*; $\neq$: different; chrm.: chromosome; int.: intergenic.

2.4 Discussion

The kinds of scenarios observed in both experiments explored in this chapter, one with real RNA-Seq data and the other with simulated NGS genomic data, provide strong evidence that *de novo* sequence assemblers create hybrid reference sequences combining different types of reads that should not be grouped together. The putative mechanism behind this is shown in Figure 2.5, and real cases, like

the ones captured by the pipeline and exemplified by Figures 2.7 and 2.9, provide support for this theory. The mechanism leads to a swap of a base in the reference sequence which, later in the mapping stage, becomes responsible for the generation of FP SNPs at the particular locus. These findings also support those of Phillippy et al. (2008) regarding SNP signatures due to misassemblies of the reference sequence.

Even though the homozygous SNP occurrences — supposed to be generated due to the reference misassembly issue explored here — were relatively few if compared to the total numbers of assembled sequences (around 1% of events on each experiment), this investigation has proved that, for most of the cases analysed, the *de novo* assembler was unable to prevent misassembly that led to the later FP SNPs. This is due to the presence of different classes of reads covering the location for which the assembler is trying to infer the consensus sequence. In the Bowman dataset experiment, the pipeline detected a range of approximately 82 to 94% of cases, out of the 473 input events, where such different classes of reads were present, depending upon the mismatch rate applied in the relaxed mappings. As expected, more classes of reads (characterised by the presence of more than one alternate allele) manifested themselves as the mappings became less stringent. This translated into a pronounced decrease in the percentages of events with only one alternate allele, which varied from 91 to 41%. For the *A. thaliana* simulated

dataset experiment, a range of approximately 44 to 47% of cases had different classes of reads involved, out of the initial 72, with the gradual increase of the relaxed mappings. The presence of a third allele (or more classes of reads) in the sequences involved in the mappings was less noticeable, as the percentages of cases with only one alternate allele and the reference one were less variable, declining from 97 to 94% with less stringent mappings.

One particular behaviour of the *de novo* assembler used to assemble the genomic simulated dataset was noticed here. As exemplified by Figures 2.16 and 2.17 and stated in subsection 2.3.2, 35 cases out of the initial 72 presented homozygous patterns in the corresponding AFG file portion of the given contig assembly. For each of these cases, the visual inspection of each read origin overlapping the SNP locus and subsequent AFG file-Velvet *Sequences* file read IDs association showed that the reads belonged to the same region of the same chromosome in each respective case. In this kind of scenario, the assembler has just got the reference base wrongly, for no apparent reason. There is certainly no evidence of paralogs or other kind of very similar sequences confounding the assembler here and, currently, there is no reasonable explanation for this "glitch" of the assembler in question. Apparently, this explains why less than 50% of the input cases were detected as having more than one class of reads taking part in the assembly.

The second stage of the pipeline aimed to shed light on the reads confounding

the assembler, and whether these are related to paralogs. This was done by BLASTing reads against databases containing sets of potential paralogous sequences. For the Bowman dataset, the database was composed of 22,651 sequences corresponding to non-redundant Haruna Nijo barley FLcDNAs. Results showed that a range of 29 to 35% of cases presenting different classes of reads (containing either the reference or the alternate alleles in the reads overlapping the SNP site; Table 2.1) had scored a 'perfect' match with a given FLcDNA region of the database file. Even though the figure is low, as mentioned in subsection 2.2.2, it was presumed that not all Bowman reads would necessarily have perfect BLAST hits in the Haruna Nijo FLcDNA database due to the inherent variation between these two cultivars of barley. Two subsequent experiments examined the effect of varying the number of sequences in the BLAST database. The number of BLAST hits for mixed (i.e. reference/alternate allele) sets of reads covering FP SNP sites increased in linear fashion with the number of FLcDNAs available in the database (Tables 2.2 and 2.3). The graph in Figure 2.18 illustrates this. Considering the increase in the availability of the barley Haruna Nijo FLcDNA sequences, on both "RANDOM" and "NON-RANDOM" experiments, the percentages of events with hits for reads overlapping FP SNP sites and belonging to different classes were close to 30% for an assumed complete database file.

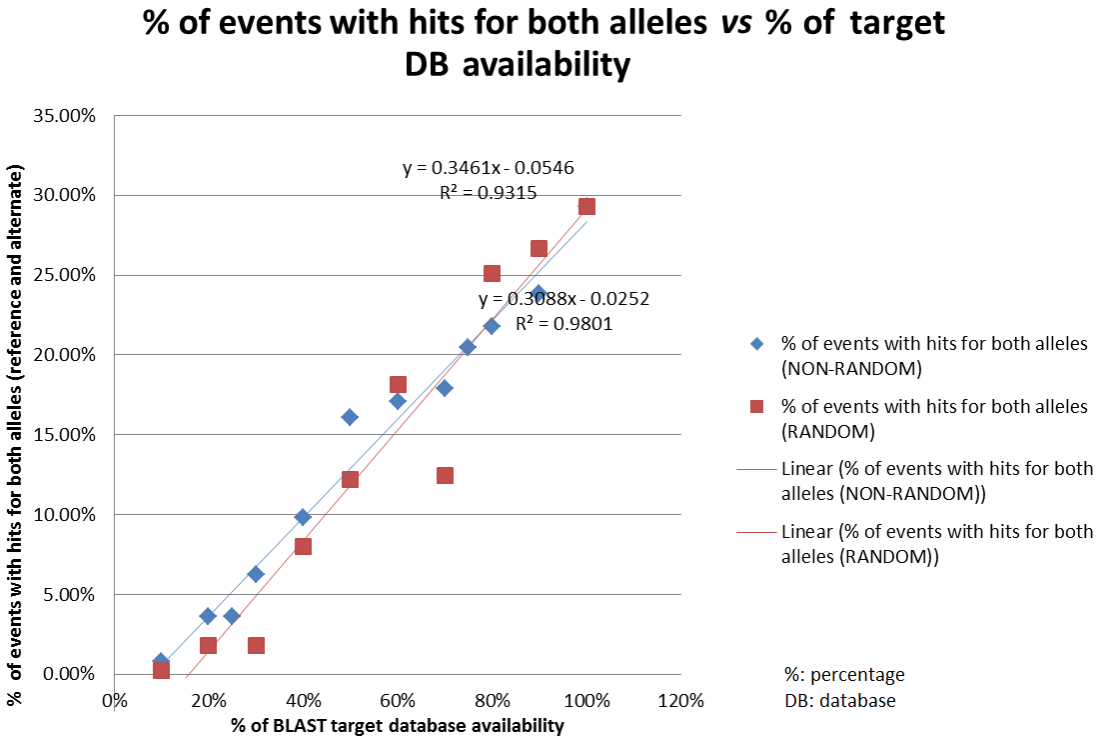

Figure 2.18: Percentages of events with BLAST hits, in the increasing databases of Haruna Nijo FLcDNAs ("RANDOM" and "NON-RANDOM" experiments), for reads of the barley cultivar Bowman dataset traversing FP SNP sites and containing either the reference or the alternate alleles. The percentages of such events were close to 30% for an assumed complete database file.

Thus, assuming that the actual barley Haruna Nijo FLcDNA file is incomplete in terms of representing the barley cultivar Bowman dataset — for instance, Matsumoto et al. (2011) state that their Haruna Nijo dataset could represent 47 to 59% of the total number of genes present in barley —, if the scenarios are extrapolated towards a continuing increase in both databases' sizes ("RANDOM" and "NON-RANDOM" experiments), the following trend emerges (Figure 2.19), given a trendline forecast of 2.35 periods.

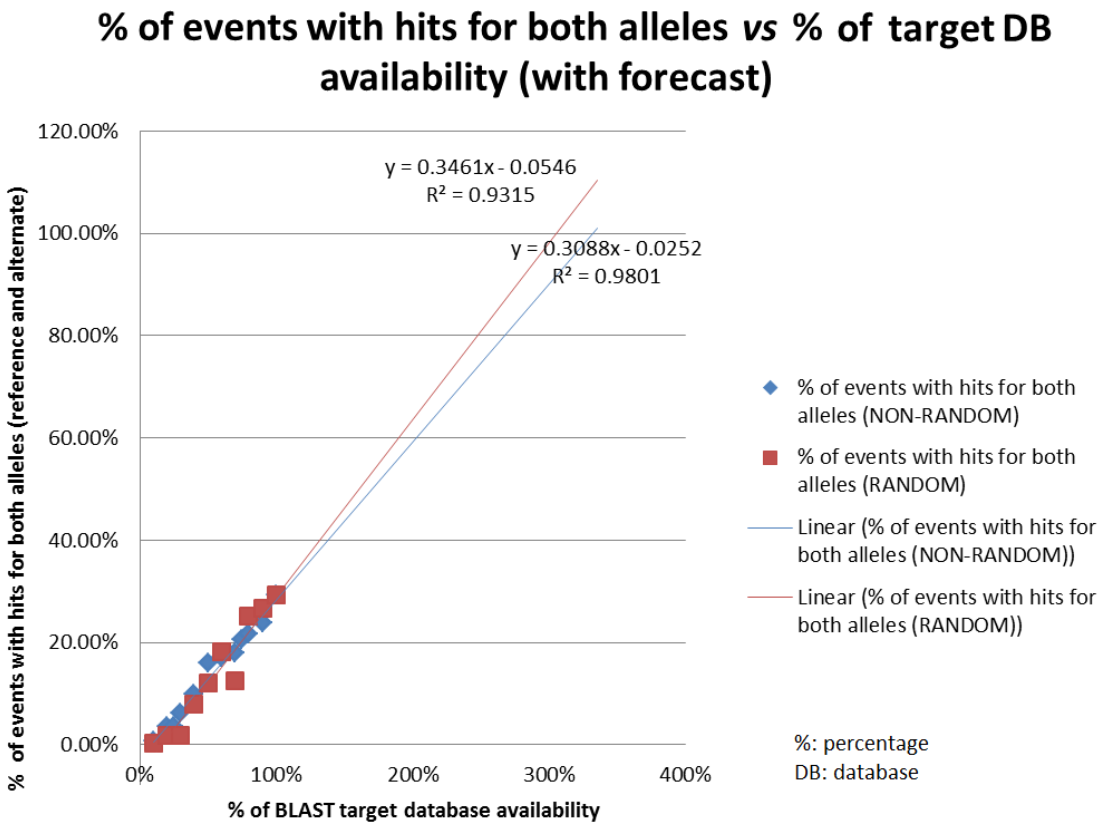

Figure 2.19: Percentages of events with BLAST hits, considering a continuous increase in the sizes of the databases of Haruna Nijo FLcDNAs ("RANDOM" and "NON-RANDOM" experiments), for reads of the barley cultivar Bowman dataset traversing FP SNP sites and containing either the reference or the alternate alleles. The estimate is that, with a 3-fold increase in the actual database file, the percentage of events with hits for different classes of reads would achieve 100%.

Thus, to obtain 100% of events with BLAST hits for reads traversing the FP SNP sites and containing the two types of alleles (reference and alternate), the actual full length cDNA database/file would need to be roughly three times larger than it is. Most probably, then, the majority of unexplained misassemblies (and related homozygous FP SNP occurrences) is due to a lack of suitable paralogs in the BLAST target database.

Another potential cause of misassemblies like the ones observed here could be the existence of distinct haplotypes at SNP sites (Cao et al., 2015; Snyder et al., 2015). Figure 2.20 is a schematic of the same misassembly mechanism due to fundamentally the same problem.

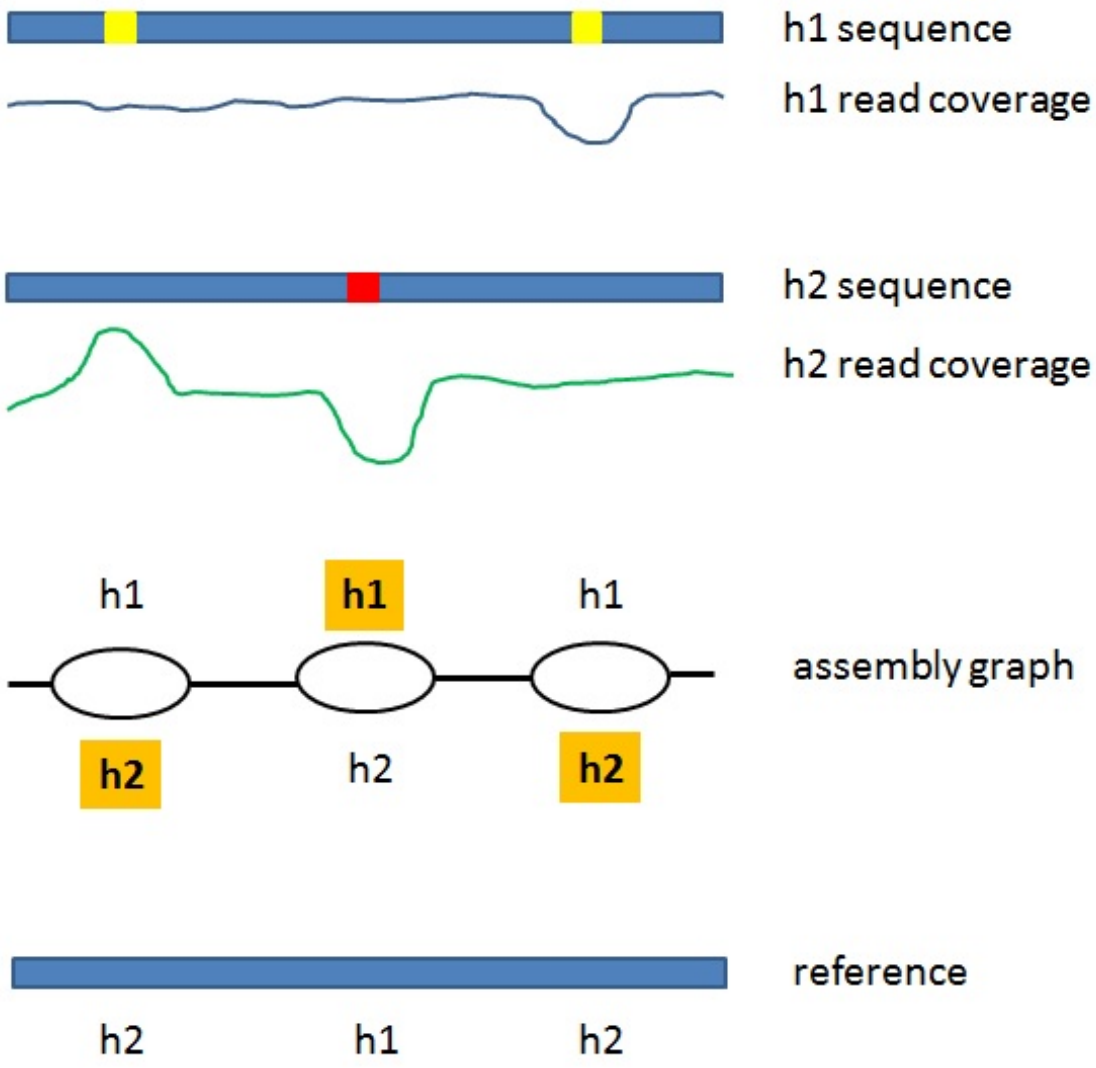

Figure 2.20: Misassembly due to different haplotypes. Schematic based on concepts described in Cao et al. (2015) and Snyder et al. (2015).

Different haplotypes h1 and h2 are shown with their respective read coverage patterns. When the assembler is trying to resolve the consensus sequence, bubbles are formed in the assembly graph. Based on the prevalence of each corresponding haplotype region, the 'reference' is inferred. A subsequent mapping of the original reads to this assembly will generate FP SNPs, just like the ones caused by misassemblies due to paralogs or other types of very similar sequences.

Regarding the *A. thaliana* dataset, subsection 2.3.2 showed that around 97% of cases detected by the pipeline as containing either the reference or alternate alleles in the reads taking part in the assembly had 'perfect' matches to different sequences of the *A. thaliana* annotation database file. Some of these cases were exemplified in Figures 2.12, 2.13, and 2.14 and, fortunately, with the available annotation from the read simulator, the different original locations of the reads taking part in the assembly can be resolved. The analysis of the corresponding AFG files provides ultimate tracking of what happens during assembly, and has pinpointed the very reads that caused the misassemblies observed. Reads originating from different chromosomes or different regions of the same chromosome provide the most direct proof of the misassembly mechanism. In fact, this revised approach — usage of simulated, haploid, and error-free reads based on *A. thaliana* model organism genomic sequence; a *de novo* genome assembler capable of generating the AFG file; and reads retrieved from such AFG file for each contig — provides a much more

tightly controlled scenario for proving that reference misassembly causes FP SNPs. For instance, in the first experiment, although Bowman cultivar was assumed as a homozygous organism, it is possible that some degree of heterozygosity is still present. Thus, with more relaxed mappings, the 'revealed' patterns of heterozygosity in the covering reads could, in fact, be genuine in some cases. Furthermore, although the best effort was made in terms of having the reads originally used by Trinity retrieved and mapped with relaxed settings, in an attempt to reveal the reads taking part in the assembly, such approach will never substitute the fine detail provided by the AFG file.

Tables A.5 and 2.6 compile information about the potential read origins, in terms of genomic locations and features, related to the respective alternate and reference alleles involved in the events. In summary, approximately 44% of the 34 events listed in Table A.5 are due to transposable element genes. Repeat sequences from intergenic regions respond for other approximate 44% of the cases. Based on Table 2.6, without considering the case with no hit for the reference allele, approximately 51% of the events are related to read origins exclusively in different chromosomes, around 18% are related to origins exclusively in different regions of the same chromosome — reads manually inspected in the downstream analysis and identified as have been originated in the same chromosome but having non-intersecting boundaries if compared to each other in terms of their respective

read labels' region ranges —, while other approximate 30% have potential origins in different chromosomes or different regions of the same chromosome.

2.5 Conclusions

The results obtained with the pipelines developed here have shown that, due to reference misassembly by the *de novo* assembler, different classes of reads can contribute to the generation of FP SNPs. This is due, at least in a probable substantial proportion of cases, to the presence of paralogs (gene duplicates), as detailed in the Bowman cultivar experiment. As observed in the *A. thaliana* experiment, sequences like transposable element genes and repeats in intergenic regions can also cause misassemblies and the consequent FP SNPs.

Such kind of new findings could be used to improve assembly tools (e.g. AFG-like files, which keep track of the assemblies and, unfortunately, are not available in some assemblers, could be incorporated in newer software versions, further allowing the type of assembly tracking/screening carried out here regarding the inspection of different classes of reads eventually taking part in and confounding the assembly).

Chapter 3

False positive SNP generation due to read mismapping

3.1 Introduction

In the experimental set up applied in Golicz's study, the same reads were used for both the *de novo* assembly and the mapping stages. In that scenario, the expectation was that any SNPs arising would be heterozygous. Apart from sequencing errors, which could act as drivers behind some such SNP events with the expected pattern, another cause suggested for that would be the cross-mapping of reads originally belonging to similar regions in the genome.

Taking advantage of the *all* and "unique" mapping modes of the Bowtie 1 tool, M. Bayer further explored this idea by simulating the occurrence of a cross-mapping event (unpublished material) and the consequent generation of FP SNPs. As per his general observations, depending on the combination of some parameters, Bowtie can be instructed to suppress all the alignments that report more than

one valid alignments and map only those reads that are unambiguously mappable (the "unique" mode). Usually, in this mode of operation, all reads that map to repeats and conserved regions in paralogs are lost. On the other hand, Bowtie can also be configured to find and report all valid read alignments (the *all* mode). In this mode, the tool maps the reads to all locations where they can be "legally" mapped (i.e. observing the mismatch parameter). Technically, a read can only ever have a single origin in the genome and therefore should only ever get mapped once, but the multi-mapping mode can be used as an option, for example, in quantitative expression studies when dealing with very low-level expression. In such a scenario, if only a single read representing a gene is present and it can map equally well to more than one gene/location (e.g. members of the same gene family), one of the following can be done: (i) either map the read once only (thereby creating false negatives for all but one location), or (ii) allow the read to map to all possible locations (thereby overinflating the expression level). The latter option, although not ideal, may be an acceptable compromise depending upon the research objective. A basic visualisation example of these different Bowtie modes is illustrated in Figure 3.1.

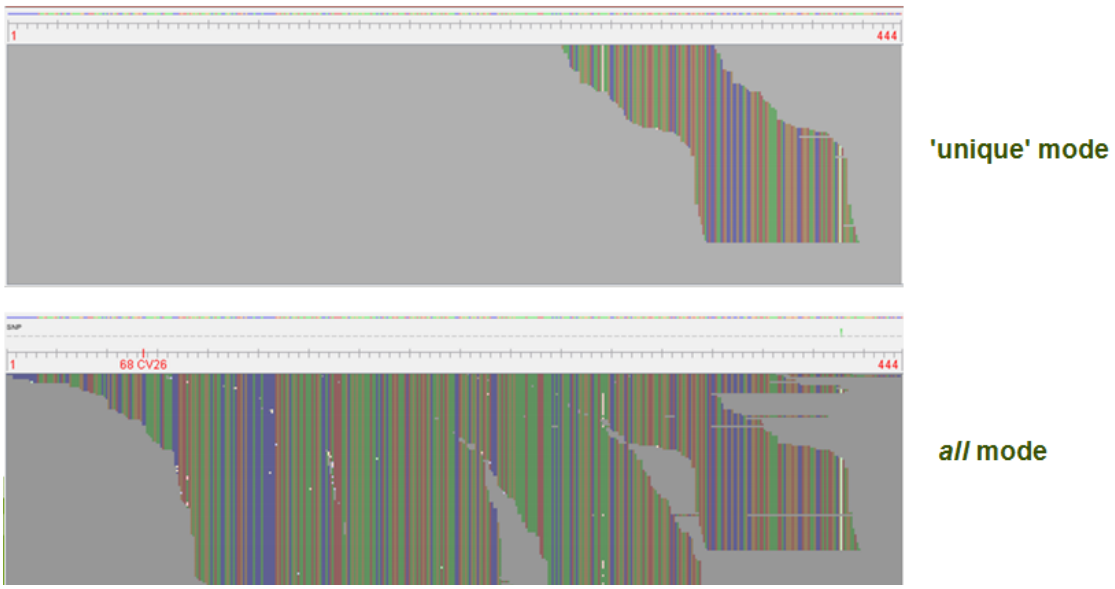

Figure 3.1: Comparison between the Bowtie tool "unique" and *all* mapping modes visualised with the Tablet tool (Milne et al., 2010). The same set of reads and reference genome were used for both modes. "Unique" mode (top of the figure) prevents reads to get mapped to all possible locations, hence reducing the read coverage area if compared to the one obtained with the tool set to *all* mapping mode (bottom of the figure). Adapted from M. Bayer (unpublished material).

In Bayer's experiment, two sets of reference sequences were initially created *in silico*: a) a FASTA-formatted file with a single reference element comprising the sequence "ACGTA"; b) another FASTA-formatted file with two reference elements, the first containing the same sequence mentioned above ("ACGTA") and the second one comprising the sequence "ACCTA". The idea here was to emulate two different conditions in terms of the reference sequences availability. In the latter file, two references would be present while, in the former, only one of the sequences would be available.

Then, a FASTQ-formatted file was built comprising three reads matching the

sequence “ACGTA” plus three other reads matching the sequence “ACCTA”. With this file, both scenarios of reference sequence availability (mentioned in the previous paragraph) can be tested in the presence of different mapping modes. Following the generation of input files, three distinct Bowtie mapping modes were evaluated: *all* mode; “unique” mode; and “unique” mode plus flags `--best --strata`. Reads were mapped against the single-sequence reference file and then to the two-sequence reference. All the mappings were performed allowing 1 mismatch to emulate a typical SNP discovery protocol. The results are summarised in Figure 3.2.

When two reference sequences are available (Figure 3.2(A)), with *all* mapping mode, all six reads are considered valid to be aligned to each reference sequence. Perfect match alignments and alignments with a mismatch are all acceptable due to the mapping mode chosen. When the “unique” mapping mode is applied, no valid alignments are reported, as each read has more than one reportable alignment — the perfect-matching read (0 mismatch) will align to its corresponding reference sequence and the other type of read (with 1 mismatch) will map to that reference as well. Finally, with the `-m 1 --best --strata` switches applied, a “weaker form of uniqueness” is applied by the tool and only the “best alignment stratum” is reported for each type of read. The second best stratum for each read (1 mismatch with the reference sequence) is not reported in this scenario.

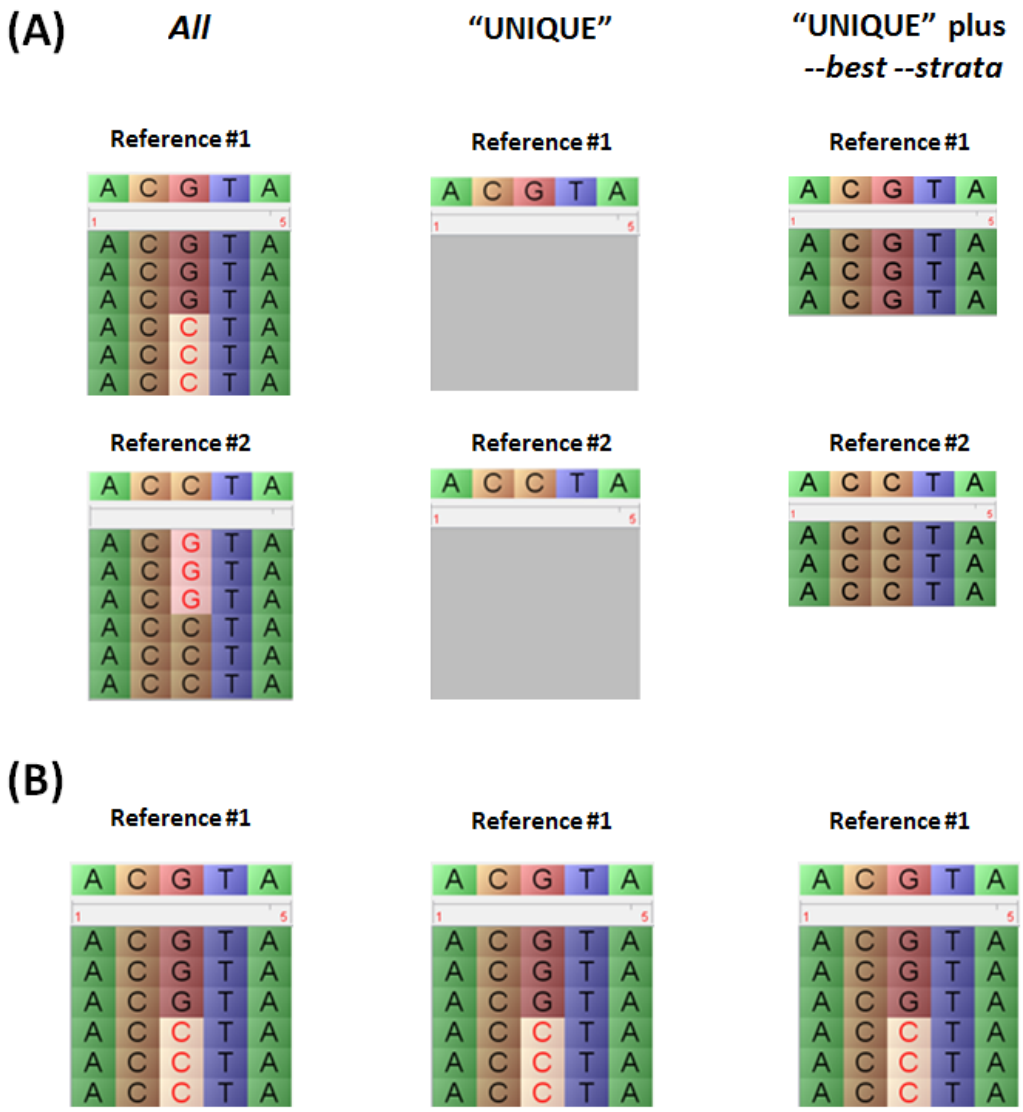

Figure 3.2: Tablet screenshots illustrating the comparison between the Bowtie mapping modes *all* (left), "unique" (middle), and "unique" plus flags `--best --strata` (right). In (A), two very similar read sets are mapped against two corresponding reference sequences with 1 mismatch allowed. In such scenario, both read sets can be mapped to their respective correct reference sequence without mismatches. There is no cross-mapping of reads as long as the "unique" mapping mode is used in conjunction with the `--best --strata` flags. In (B), the two similar read sets are mapped against a single reference sequence only, also with 1 mismatch allowed. In this scenario, only one of the read sets matches the reference sequence perfectly while the second class of reads is mismapped, thereby creating a FP SNP. Adapted from M. Bayer (unpublished material).

As per the Bowtie manual (Johns Hopkins University, 2009), in Figure 3.2(B), specifying the *all* mapping mode with the `-a` switch instructs Bowtie to report all valid alignments. Reads with the sequence "ACGTA" perfectly match the single reference sequence available and are all aligned to it. Since 1 mismatch is being allowed, the reads with the sequence "ACCTA" are "legally" mappable as well. When the "unique" mapping mode is applied, the `-m 1` switch combination tells Bowtie to refrain from reporting any alignments for reads having more than one reportable alignment. Since there is only one reportable alignment per read, all reads are, again, mapped. The same happens even when the flag `--strata` is applied (along with its mandatory counterpart flag `--best`) in association with the "unique" mode `-m 1` switch. One valid reportable alignment still exists per read and the result is the same as with the other two mapping modes.

As an addendum, "stratum" means a set of reads matching the reference with a given number of mismatches (Figure 3.3). So, with `--strata`, alignments are ordered by number of mismatches. The `--best` flag ensures that only reads in best stratum are mapped. As one conclusion of Bayer's experiment, by assuring that only reads in best stratum are mapped, the `--best --strata` combination has the potential to reduce the risk of cross-mapping between related reference sequences and, consequently, the number of FP SNPs.

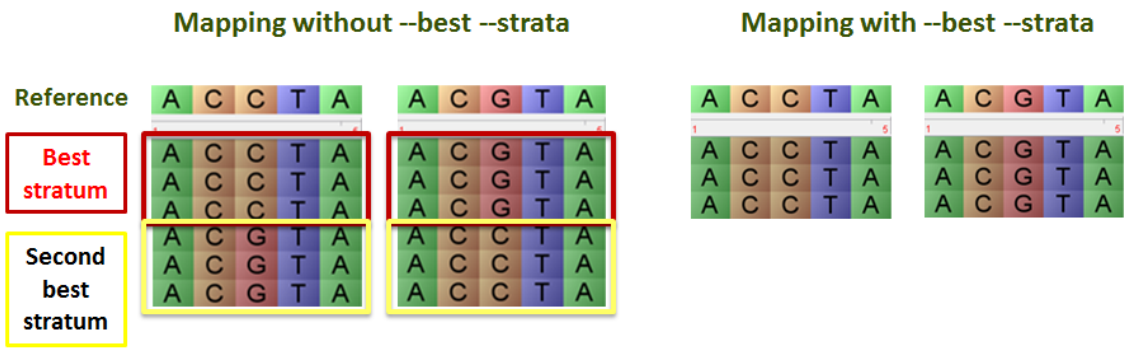

Figure 3.3: Tablet screenshots illustrating the comparison between the Bowtie tool behaviour with and without the `--best --strata` flags applied. The second best alignment stratum for each read is not reported at all when the flags are used. Adapted from M. Bayer (unpublished material).

The relationship between mismapping and FP SNPs has also been highlighted by Milne et al. (2013b) as one of the NGS data visualisation examples provided by the Tablet graphical viewer (Milne et al., 2010, 2013a). The tool can be used to identify mismapping and misassembly errors which can potentially generate FP SNPs or erroneous splice junctions. In the example of Figure 3.4, transcripts that have been *de novo* assembled had the RNA-Seq reads mapped onto them using the Bowtie mapping tool.

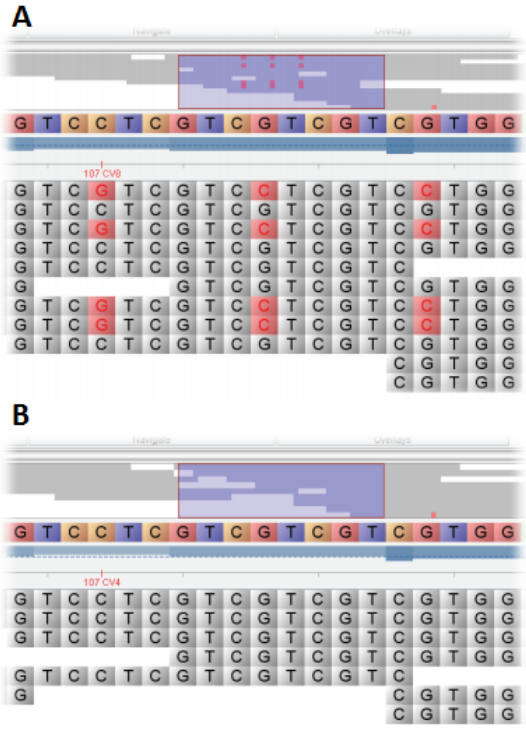

Figure 3.4: Visualisation of mismapped reads with Tablet tool. (A) Ambiguously mappable reads mapped to all their possible locations visualised with the Tablet tool in the Bowtie *all* mapping mode. Three FP SNPs generated. (B) No FP SNPs generated when read cross-mapping is suppressed with Bowtie `--best --strata` switches. Adapted from Milne et al. (2013b).

In Figure 3.4(A), the *all* mode is used and ambiguously mappable reads are mapped to all of their possible locations, resulting in cross-mapping of reads that belong to another, very similar, transcript. In this case, this results in three FP SNPs during the SNP discovery stage. In Figure 3.4(B), Bowtie's `--best --strata` switches are set and suppress the cross-mapping, by allowing mapping only to the best fit single location (lowest number of mismatches). Since the mismatch rate of the reads that were mismapped originally is higher than the

specified threshold, they are prevented from mapping and no FP SNP arises.

It is important to recall from Bayer's experiment that, in a scenario where a genuine SNP is not present within the reads, FP SNPs occur, in any combination of mapping mode, when only a single reference is available. Furthermore, even with two reference sequences available, a FP SNP would still arise with the mapper being run in *all* mode. In each of these scenarios, the FP SNPs are caused by mismapping.

Thus, this chapter then focuses on the phenomenon of heterozygous FP SNPs being generated by read cross-mapping, irrespective of mapping mode. In Chapter 2, the effect of reads from very similar genomic regions on *de novo* assembly and the consequent generation of false positive SNPs was investigated. Here, the main point of investigation is to measure the impact of these very similar reads on the mapper and how the latter can be "confounded" by them, inducing further FP SNPs. This includes cases where the corresponding reference sequence is absent from the *de novo* assembly (Figure 3.5).

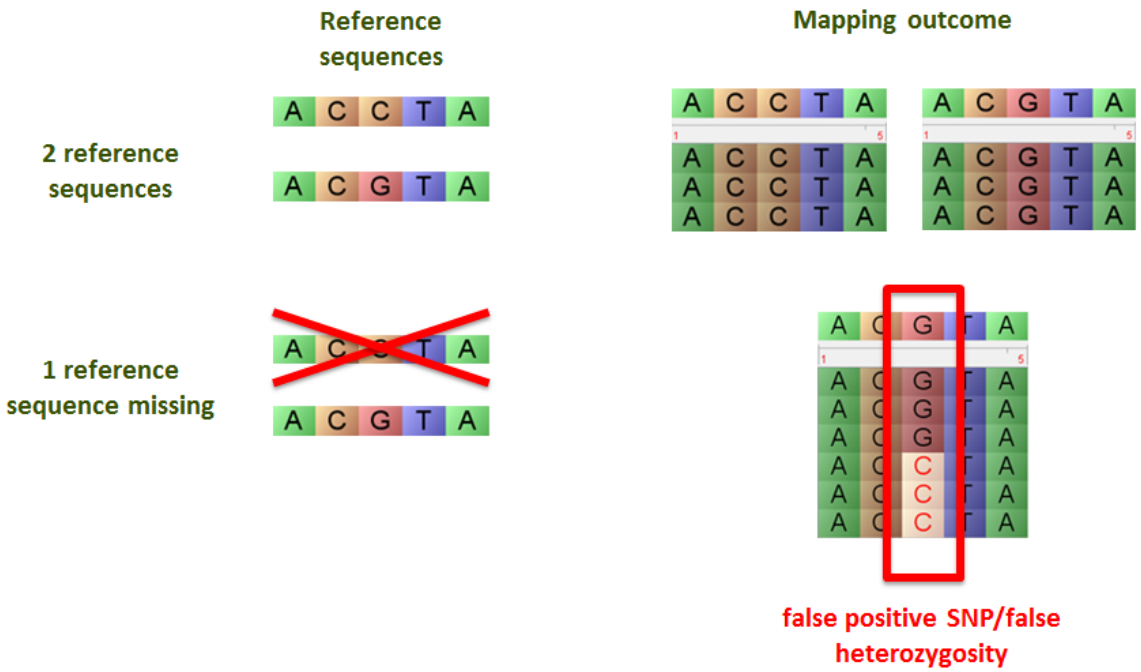

Figure 3.5: A FP SNP/false heterozygosity scenario exemplified with Tablet screenshots. On the top part of the figure, the mapping outcome when two reference sequences are available. On the bottom part of the figure, a FP SNP being generated by cross-mapping of very similar reads aligned to the only reference sequence available. Adapted from M. Bayer (unpublished material).

Thus, the following questions were the motivation for this chapter's investigation:

- Can read cross-mapping produce FP SNPs in a mapping-based SNP discovery approach when the reference sequence is poorly assembled (e.g. contains gaps)?

- If so, what kind of genomic features are more prone for inducing cross-mappings?

3.2 Methods

3.2.1 Datasets used

The 150 bp paired-ended simulated read dataset, sampled from the *Arabidopsis thaliana* genome, as described in Chapter 2, subsection 2.3.1, item 2.3.1.1, was used for the experiment. For additional information, see Appendix A, subsection A.1.2, item A.1.2.1). The rationale behind the experiment is that, due to the read simulation model used (haploid genome, no read errors), this excludes all potential sources of SNP variants in the reads, apart from mismapping artefacts, allowing the conclusion that every observed heterozygous SNP encountered is a FP due to cross-mapping.

In order to provide the conditions typical of a non-model organism use case, the reference sequence for the subsequent read mapping stage was *de novo* assembled from this 150 bp read dataset using Velvet version 1.2.10 (Zerbino and Birney, 2008). Automated parameter tuning with VelvetOptimiser.pl script version 2.2.5 (see Victorian Bioinformatics Consortium (2012); (Zerbino, 2010)) was used to optimise results. Additional information about the assembly process can be found in Appendix B, subsection B.1.1, item B.1.1.1. Assembly statistics retrieved from the VelvetOptimiser script as well as the QUAST tool (version 2.1) (Gurevich et al., 2013) can also be seen in Appendix B, subsection B.1.1, item B.1.1.2.

Following the same approach of Chapter 2, the 150 bp read dataset was aligned

to the *de novo* assembled reference sequence with Bowtie2 version 2.2.1 set up to allow 3 mismatches to enable the subsequent SNP calling stage (see Appendix A, subsection A.1.2, item A.1.2.4, for more details about this setting). Mapping results can be found in Appendix B, subsection B.1.1, item B.1.1.3.

SNPs were then called using FreeBayes version v9.9.2-23-g7e198dc-dirty (see Appendix A, subsection A.1.2, item A.1.2.6, as a reference of the applied command and parameters). The custom classifier and tallying code (M. Bayer, unpublished material), mentioned in Chapter 2's 2.3.1 subsection, item 2.3.1.1, was used to analyse the VCF file from FreeBayes. The text file generated was filtered to retrieve only the heterozygous SNP occurrences and, to simplify the downstream analysis, multi-allelic events were discarded. This final list was used as the input for another pipeline developed to evaluate and quantify read mismapping (see Appendix B, subsection B.1.1, item B.1.1.4).

The BLAST database from Chapter 2's *de novo* genome assembly experiment (see subsection 2.3.1, item 2.3.1.1) was used to serve both as an input file for the analyser pipeline and also as a standalone resource. In this latter role, the database was queried to test whether regions containing SNPs were enriched for specific genomic features, such as intergenic regions, gene families, pseudogenes, repeats, and transposons. To compare the proportions observed with those observed in the entire genome, SNP *manifests* (SNP site plus approximately 120 bp flanking

region either side) were extracted from the *de novo* assembly, with custom Java code (M. Bayer, unpublished material) and queried against the database.

Finally, the *A. thaliana* chromosome sequences (described in more detail in item 2.3.1.1 of Chapter 2's subsection 2.3.1) were combined to serve as a control reference sequence. The read mapping, SNP calling, and annotation stages were also applied to this original genome sequence. This approach should theoretically yield no or at least fewer SNPs, as the additional complication of the *de novo* assembly is removed here, and should therefore act as a control for the experiment's assembled reference sequence. Figure 3.6 illustrates the control concept.

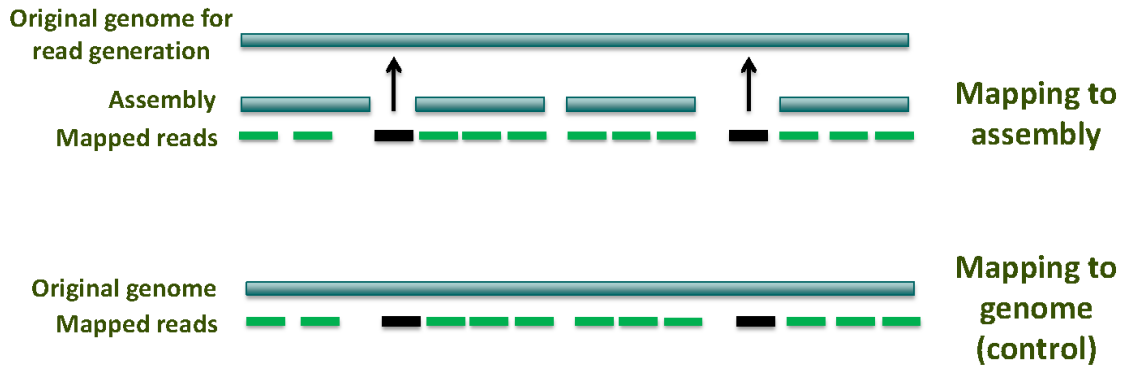

Figure 3.6: Control conceptualised. The reads indicated by arrows cannot be mapped to their original positions in the *de novo* reference genome assembly, due to gaps or misassembly, and may therefore map to the wrong location, which potentially results in FP SNPs. In the control mapping to the complete genome, the same reads can map back correctly to their original positions. Adapted from Ribeiro et al. (2015).

3.2.2 Software implementation and use

A pipeline was developed to quantify instances where mismapped reads cause SNPs, taking advantage of the read origin information generated by the read

simulator (exemplified in Figure 3.7). The code workflow is shown in Figure 3.8.

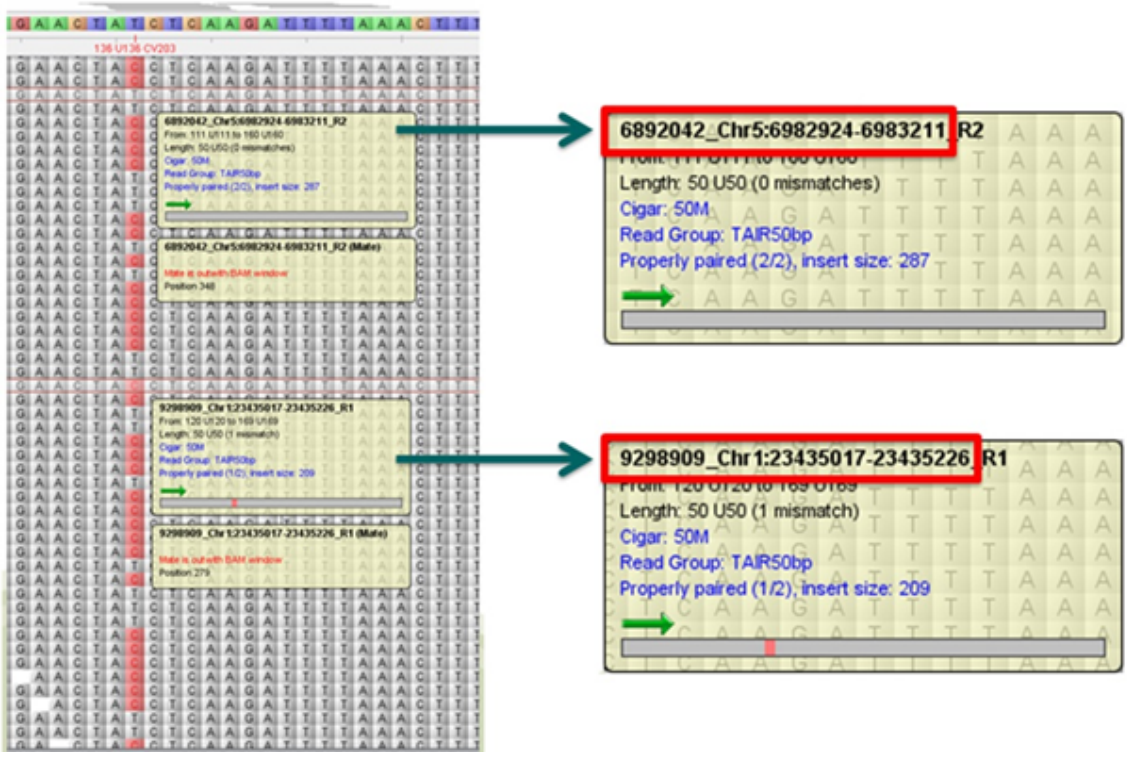

Figure 3.7: An example of mismapping identified by the read origin information available from the read simulator. Screenshots, from the Tablet assembly viewer, show a FP SNP caused by read mismapping. Sherman read labels are outlined in red. The read with the alternate allele belongs to a different chromosome when compared to the ones with the reference allele.

The code was written in Java and used resources like the Picard API, the SAMtools version 0.1.18 suite (Li et al., 2009a), and the local alignment search tool BLASTN (Altschul et al., 1990). The program uses the list of heterozygous SNPs to be analysed (from the existing tallying code), the assembly, the mapping, and the BLAST database as input files. It scans for each SNP within each contig extracting the unique overlapping (covering) reads at each SNP site. Then, it

counts the number of reads containing the same allele as the reference sequence as well as the number of reads containing alternate alleles. By querying the BLAST database, it also checks for the original position and allele, in the original genome, that corresponds to the SNP site in the contig. Based on this 'anchoring' information as well as the read labels' region ranges, it computes the percentage of mismapped reads containing the allele alternate to the corresponding allele in the genome, writing the output to a results file. The read labels' region ranges are also used to determine whether the reads originally belong to a different chromosome or different region in the same chromosome, now in an automated implementation of the manual approach mentioned in Chapter 2's *A. thaliana* experiment (Section 2.4). The code also covers the potential situation where a disagreement occurs between the allele in the assembled contig and the corresponding position in the original genome sequence (based on the BLASTN analysis; i.e. single base misassembly in the contig). In these situations, the reads containing the allele observed in the contig are counted as the mismatched ones, as they contain the allele that differs from the original genome sequence.

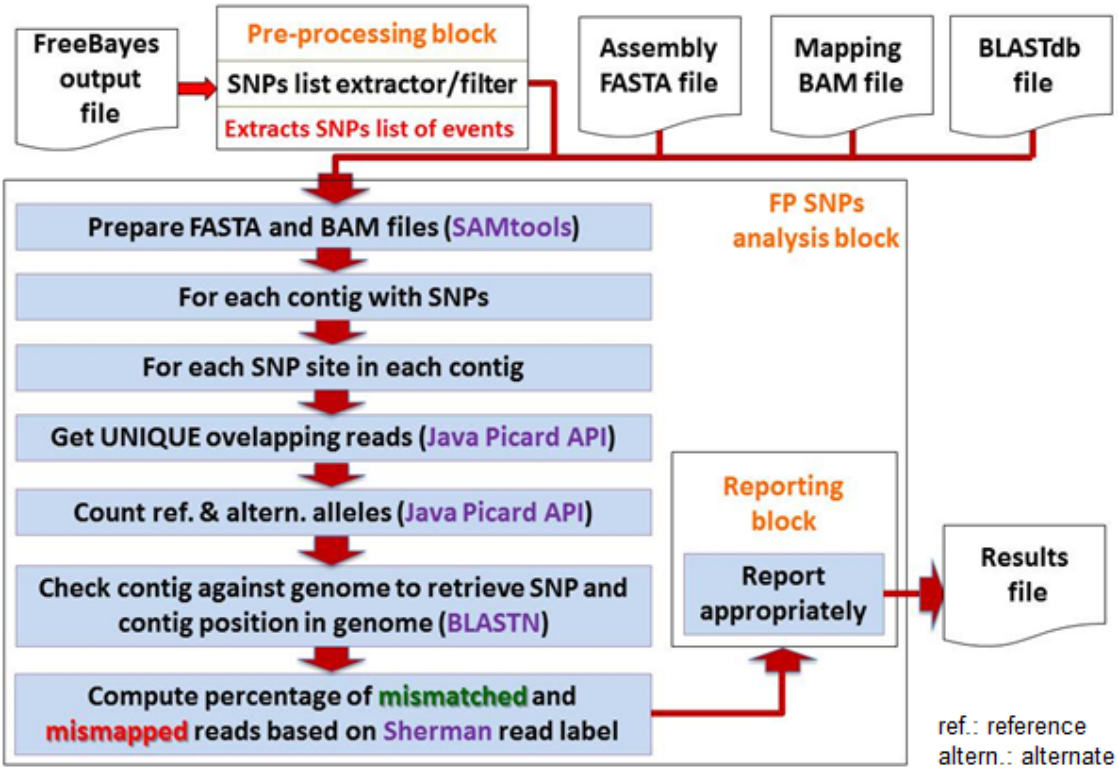

Figure 3.8: Workflow of the mismapped read quantification code. See Appendix B, subsection B.1.3, for the link to the source code.

Figure 3.9 illustrates the experimental design.

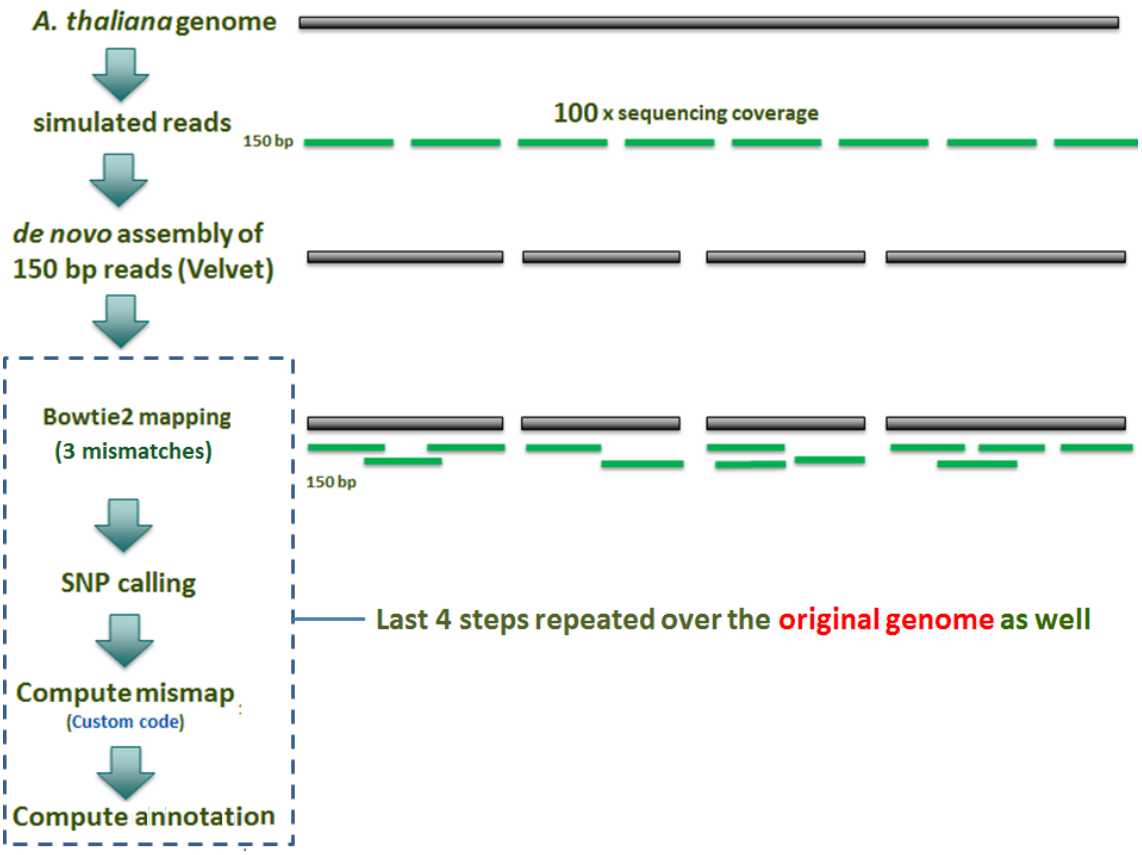

Figure 3.9: Design of the read mismapping proof-of-concept experiment. The *A. thaliana* genome was used to generate simulated 150 bp paired-end reads. A *de novo* assembly was computed from the read dataset and served as reference for a mapping of the same reads. SNP calling was carried out and the results were analysed with custom code to detect whether the mismatched reads causing the SNPs were due to read mismapping. SNP annotation was performed to detect enrichment for particular types of genomic features at SNP positions. The read dataset was also mapped to the original genome and the SNP calling, mismap quantification, and SNP annotation steps were repeated to act as a control for the experiment.

3.3 Results

The SNP calling stage reported 44,638 SNP events when the reads were mapped to the *de novo* assembly, while 858 occurrences were found when aligning to the

control genome. Read alignment rates varied from 99.41% to 100%, respectively. The rate of occurrence of mismapped reads among reads with alternate alleles at SNP locations was 95.94% in the *de novo* genome assembly scenario and 98.45% in the control genome.

Regions associated with FP SNPs were strongly enriched for transposable element genes: 37.67% in the *de novo* assembly mapping and 16.39% in the control genome mapping, compared to 6.01% occurrence of such elements in the original genome annotation (Figure 3.10 and Table 3.1).

Table 3.1: Number of occurrences retrieved by the used annotation approach

Categories defined for SNP characterisation	BLAST database entries	SNP entries (*de novo* assembly)	SNP entries (control genome)
Family	10,530	1,104	55
Intergenic	31,342	26,761	649
Other CDS	13,115	926	30
Pseudogene	876	408	8
Repeat	1,410	90	0
Reverse transcriptase	24	3	0
Specific transposon / retrotransposon	20	0	0
Transposable element gene	3,900	17,952	149
Transposase	14	7	0
Unknown protein	3,713	409	18
Totals	64,944	47,660	909

The genomic distributions of the annotated SNPs are illustrated in Figures 3.11 and 3.12, based on the original genomic positions of the FP SNP sites retrieved by the pipeline. There was generally a higher prevalence of FP SNPs in the central parts of chromosomes (The Arabidopsis Genome Initiative, 2000).

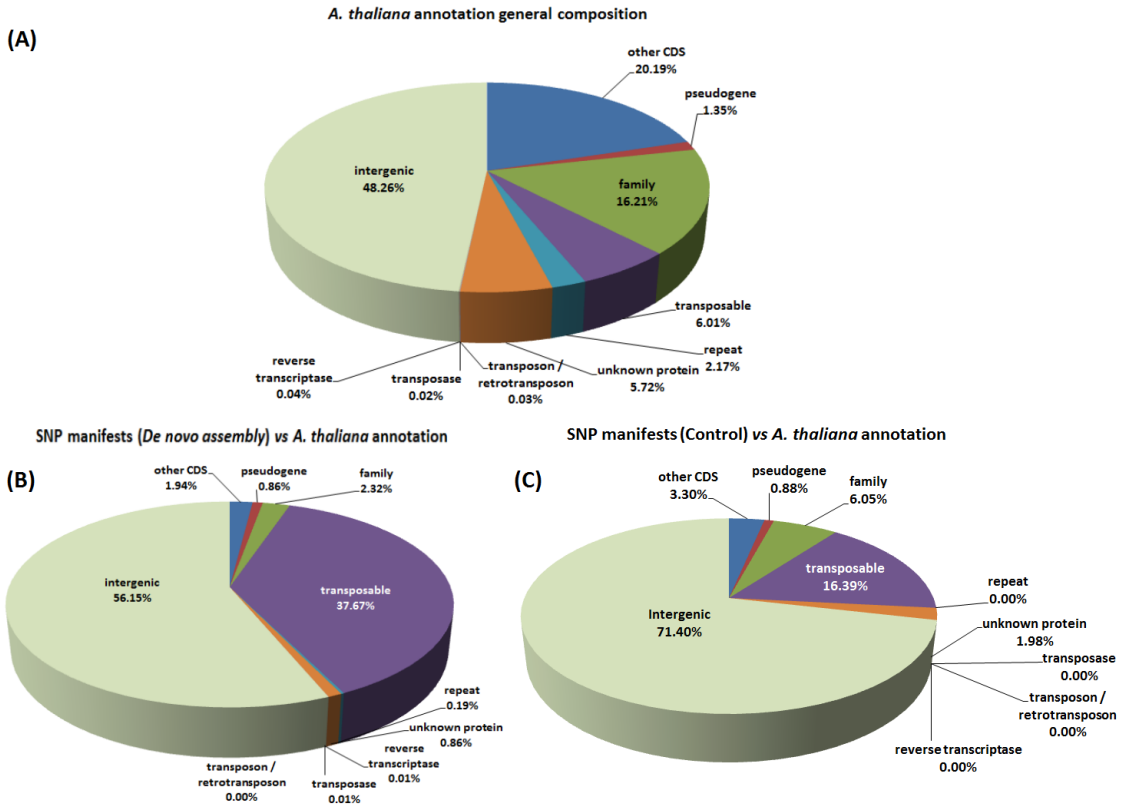

Figure 3.10: Read mismapping experiment annotation results. The *Arabidopsis thaliana* annotation (A) is compared with the BLAST-based annotation results for the SNP manifests retrieved in the experiment from mapping: (B) to the *de novo* assembly; (C) and the control genome.

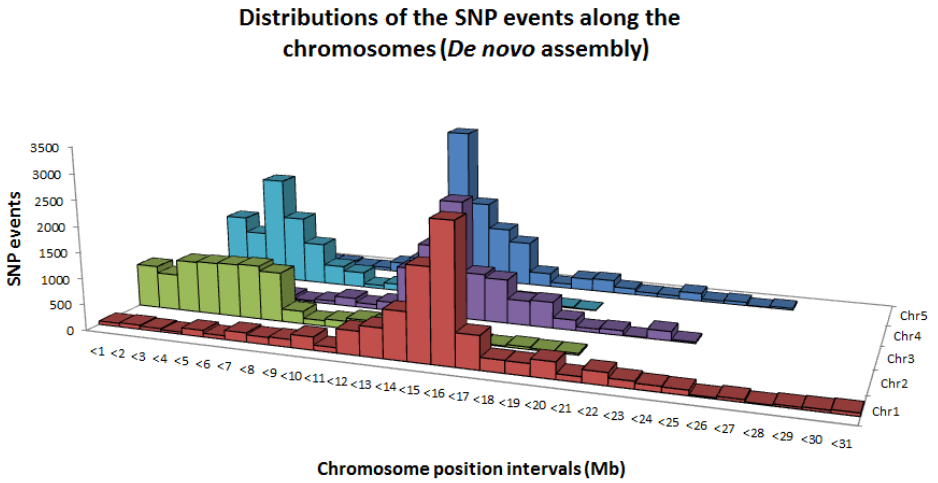

Figure 3.11: FP SNP sites genomic locations (*de novo* assembly). Plot of the distributions of FP SNP sites, by chromosome, from the mapping to the *de novo* assembly. Genomic locations are shown on the x axis divided in intervals of up to 1 mega base pairs (only upper limits depicted for simplicity). FP SNP counts are shown on the y axis. See Appendix B, subsection B.1.2, for the link to the respective table used for plotting.

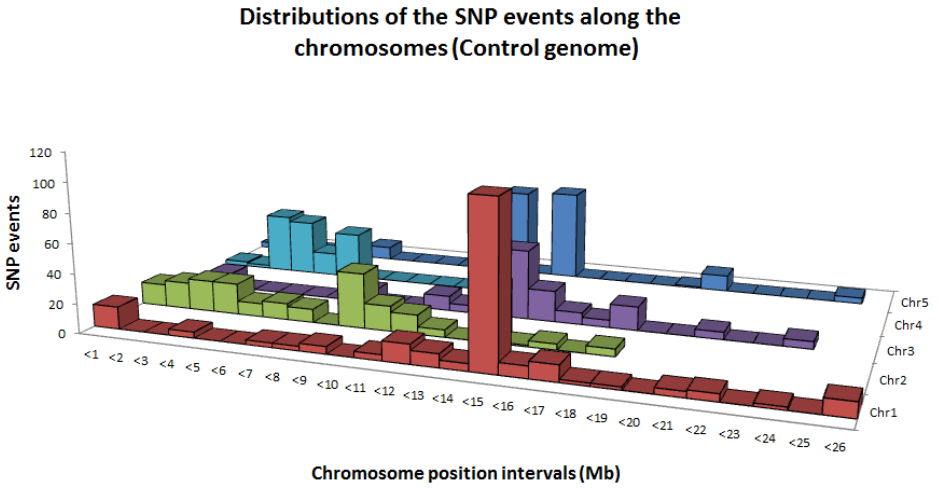

Figure 3.12: FP SNP sites genomic locations (control genome). Plot of the distributions of FP SNP sites, by chromosome, from the mapping to the control genome. Genomic locations are shown on the x axis divided in intervals of up to 1 mega base pairs (only upper limits depicted for simplicity). FP SNP counts are shown on the y axis. See Appendix B, subsection B.1.2, for the link to the respective table used for plotting.

3.4 Discussion

In this chapter's investigation, a simulation of a mapping-based SNP discovery workflow, typical of usage in non-model organisms, was carried out to test whether misassembly of the reference sequence creates the conditions which lead to read mismapping and consequently FP SNPs. Reads were mapped against the published genome of *A. thaliana* as well as a *de novo* assembly derived from those reads. In these circumstances, reads can be mismapped if their site of origin is not available. This can lead to mismatches with the reference sequence, producing FP SNPs.

This is somewhat similar to the phenomenon observed in Bayer's exploration of Bowtie mapping modes, mentioned at the beginning of the chapter. This kind of outcome is relevant, for example, to SNP discovery projects where a well assembled and curated reference sequence is not available, and a *de novo* computed reference is created to serve as the basis for the subsequent mapping and variant calling stages. In fact, the reference sequences of most sequenced organisms are classified as a "permanent draft" (JGI, 1997), and have undergone little or no manual curation following the primary assembly stage. Typically, the resulting genome sequences are fragmented and incomplete, with significant numbers of misassemblies, and often form the basis for applied work, e.g. the development of molecular markers for breeding purposes (Kumar et al., 2012). All of those mentioned imperfections may subsequently cause the type of read mismapping

observed here and, consequently, FP SNPs. In my study, a very small number of mismapping-associated FP SNPs also occurred in the mapping against the original genome, but this was approximately 52-fold lower than the discovery rate when using the *de novo* genome assembly. Those events were most probably provoked by genuine cross-mapping of very similar reads but their lower numbers reinforce the fact that the usage of a better reference sequence results in less false positives.

The read origin information available from the read simulator helped to demonstrate that almost every read (approximately 96%) with an alternate allele at SNP locations was mismapped. This demonstrates clearly that mismapping may cause FP SNPs, under certain conditions. The work of Li (2014), for example, sheds light on erroneous alignments in low-complexity regions and incompleteness of the reference genome as two major sources of errors. Here, as detected by the analyser pipeline, mismapping was mostly a consequence of the *de novo* assembly, which implies that many true read origins were not available in the reference sequence due to misassembly or non-assembly. The QUAST analysis also confirmed that the *de novo* assembly was incomplete and contained misassemblies.

The tools chosen here for assembly, mapping, and variant discovery — Velvet, Bowtie2, and FreeBayes — are all very widely used. They also incorporate robust methods commonly associated to the NGS mapping-based SNP calling workflow, respectively: a de Bruijn graph solution for the genome assembly stage, the

Burrows-Wheeler Transform technique in the read alignment stage, and a Bayesian haplotype-based polymorphism discovery and genotyping for the SNP calling stage. Regardless of that, it is conceivable that the numbers of FP SNPs observed here might have been slightly different with the use of other tools or parameters. For instance, it is important to notice that FreeBayes was used here in a "naive" running mode (Garrison, 2014) and that, potentially, the use of additional filtering would have reduced the FP SNP numbers observed. Even so, it is unlikely that FP SNPs caused by read mismapping would have been completely prevented by just applying filters, reinforcing the confidence that the pattern observed here is real.

To validate the read mismapping concept in the current study, the choice was for the simple design of a Sanger-sequence based, high-grade reference sequence *versus* a non-curated, short-read based reference containing substantial numbers of misassemblies whilst lacking other regions completely. It is certain that investing additional resources to produce a high quality assembly would result in less mismapping and hence fewer FP SNPs, just like the results obtained with the mapping to the control *A. thaliana* assembly. This sequence, for instance, has had decades of effort invested into it. Nevertheless, potential future work to further explore the topic might include generating several assemblies for mapping reads onto, each made from reads of different lengths and/or using different fragment

sizes for the paired-end reads.

The observed enrichment for transposable element sequences in regions containing FP SNPs — approximately 38% for the mapping to the *de novo* assembly — was corroborated by reporting the FP SNP sites' original genomic positions. A large proportion of FP SNPs were located in the pericentromeric regions of the chromosomes, where such repetitive sequences are prevalent (The Arabidopsis Genome Initiative, 2000; Baker et al., 2014) (Figure 3.13). This leads to the conclusion that misassembly of repeats in the *de novo* assembly computation was the prime cause for generating FP SNPs due to read mismapping in this experiment.

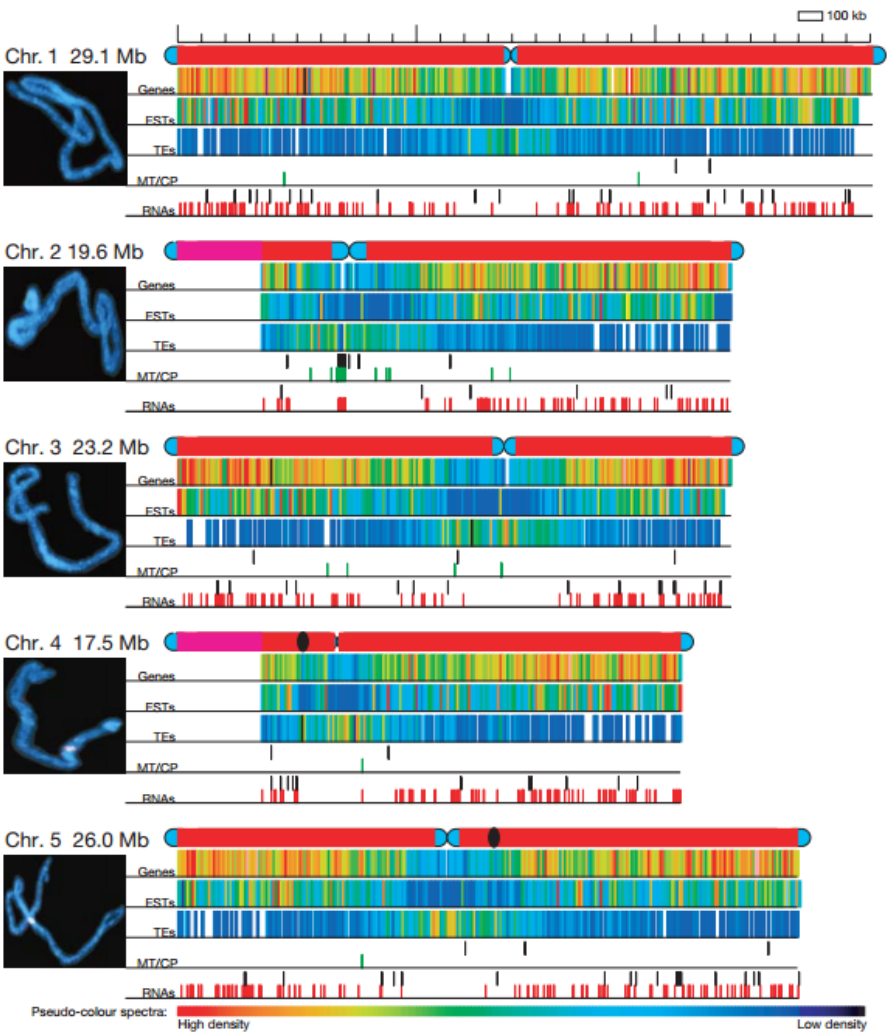

Figure 3.13: Representation of the *A. thaliana* chromosomes. As per the The Arabidopsis Genome Initiative (2000) work, each chromosome is represented as a coloured bar where sequenced portions are red, telomeric and centromeric regions are light blue, heterochromatic knobs are black, and the rDNA repeat regions are magenta. The frequency of features received pseudo-color assignments, from red (high density) to dark blue (low density). Feature densities represented: 'Genes' (38 per 100 kb to 1 gene per 100 kb), 'ESTs' (expressed sequenced tag matches ranging from more than 200 per 100 kb to 1 per 100 kb), 'TEs' (transposable element densities ranging from 33 per 100 kb to 1 per 100 kb). 'MT/CP' (mitochondrial and chloroplast insertions) were assigned black and green tick marks, respectively, and 'RNAs' (transfer RNAs and small nucleolar RNAs) were assigned black and red tick marks, respectively. Adapted from Figure 1 (The Arabidopsis Genome Initiative, 2000), by permission from Macmillan Publishers Ltd: Nature, copyright 2000.

3.5 Conclusions

The main message from this work is that mismapping due to poorly assembled reference sequences can cause FP SNPs in potentially large numbers. This is because misassembled reference sequences create the conditions required for read mismapping (i.e. absence of reads' true origins), with transposable element sequences being particularly prone to misassembly. Mismapping of reads then can lead to FP SNPs due to mismatches at the mapped location between the (available) reference sequence and the reads.

Chapter 4

A multifactorial experiment to evaluate false positive SNP generation due to read mismapping

Disclaimer

This chapter formed the basis of Ribeiro et al. (2015). It presents the results of a collaboration between the author and seven other researchers: Ms Agnieszka Golicz, Dr Christine Hackett, Dr Iain Milne, Mr Gordon Stephen, Dr David Marshall, Prof. Andrew J. Flavell, and Dr Micha Bayer. As described in this thesis' Chapter 2, Ms Agnieszka Golicz carried out the initial research on FP SNPs. Dr Christine Hackett aided with the data statistical analysis. Dr Iain Milne and Mr Gordon Stephen provided a parallelisation wrapper for FreeBayes software in order to speed up this package's processing. Dr David Marshall, Prof. Andrew J. Flavell, and Dr Micha Bayer coordinated the research. I conducted the

experiments, coded most of the applied tools/scripts, and pipelined third-party software when necessary, finally analysing the data and reporting the results and findings here.

4.1 Introduction

As mentioned in Chapter 1, significant numbers of FP SNPs can arise as a result from SNP discovery in NGS data (Farrer et al., 2013; Li, 2014b), when the traditional mapping-based approach is applied. Apart from sequencing errors, which vary in pattern and rates depending on the utilised NGS platform (Nielsen et al., 2011), the following factors may play a role in this to a greater or lesser extent: reference sequence quality, mapping tool, variant calling tool, mapping stringency, read mapping quality filtering, read depth filtering, and read length. Here, I try to provide an overview about the complexities associated with each of these factors.

Previous chapters of this thesis have attempted to draw attention to the harmful effects which can be provoked due to low accuracies associated with computed reference assemblies. As highlighted in Chapter 3, for instance, misassemblies or gaps are potential drivers for read cross-mappings and, consequently, FP SNPs. In fact, Kumar et al. (2012) have argued that SNP discovery improves with better quality reference genomes. The development of more robust assembly algorithms is the subject of an ongoing international research effort. A large number of these has

been introduced during the past decade (see Section "Available Assemblers", in Wikipedia (2005), for popular examples). Most assemblers have been benchmarked under diverse experimental conditions, to test their robustness and reliability (GAGE, 2011; UC Davis Genome Center, 2011; CNAG, 2011). Even though, as detailed in Chapter 1, due to inherent complexities associated with organisms' genomes (i.e. repeats patterns which may vary between species) and technical challenges (i.e. sequencing errors, drops in coverage, short read length, etc.), to name but a few, the quest for an error-free genome assembler continues.

As pointed out in studies such as Nielsen et al. (2011) and Farrer et al. (2013), accuracy of read alignment has also been shown to play a crucial role in variant detection. Differences in bases between the reference and a newly obtained sequence, which theoretically may be interpreted as variant/SNP calls, can also be caused by misalignment of short reads (Li et al., 2009b; Altmann et al., 2012). Erroneous realignment in low-complexity regions and, as mentioned in the previous paragraph, an incomplete reference genome with respect to the sample, are also great sources of problems (Li, 2014b). As described in Chapter 1, significant effort has been invested into improving the alignment stage via different algorithmic approaches (Li and Homer, 2010; Altmann et al., 2012). Nevertheless, as it happens with *de novo* assemblies, alignment accuracy is very dependent on the software used and parameters applied, the type and size of the dataset, and

the number of incorrect base calls (Farrer et al., 2013).

In recent years, commonly used mappers and typical parameter combinations have been evaluated and compared by different studies, mostly in terms of speed, efficiency, memory consumption, and accuracy (Li and Homer, 2010; Ruffalo et al., 2011; Altmann et al., 2012; Farrer et al., 2013; Hatem et al., 2013; Shang et al., 2014). Hatem et al. (2013), for instance, stated that no single tool outperformed all others in all metrics. More recently, this field of research started to focus not only on the improvements associated with the mapping tools but also on the search for the best choice of mapper and corresponding parameter tuning based on a given project characteristics (i.e. an individualised approach). The recently released *Teaser* tool (Smolka et al., 2015) is a good example of that. All of these efforts emphasize the complexity involved in the search for obtaining good and reliable alignments.

As seen in Chapter 1 and highlighted in the work of Cantarel et al. (2014), accurate detection of SNPs is not trivial. The study's authors, for instance, claim that there is no standard protocol for detecting SNP predictions with the highest sensitivity — desirable to minimise false negative calls therefore avoiding missing true mutations — and specificity — essential to minimise false positives and consequent erroneous/costly unfruitful work. Instead, each algorithm promotes a different balance of sensitivity and specificity, either increasing the number of true

positives at the cost of being susceptible to additional false positives or decreasing these latter at the cost of missing the former ones (Cantarel et al., 2014). Clevenger et al. (2015) also point out that different variant calling programs can call different polymorphims due to the different evaluation models utilised.

Because of these issues, many of the variant calling tools have been benchmarked in some form (Liu et al., 2013; Pabinger et al., 2014; Talwalkar et al., 2014; Li, 2014b). However, as highlighted in the works of Chapman (unpublished results, (Blue Collar Bioinformatics, 2013)) and by many other researchers (O'Rawe et al., 2013; Farrer et al., 2013; Talwalkar et al., 2014; Li, 2014b; Cornish and Guda, 2015; Clevenger et al., 2015), the number of tools available, their development rate, and a relatively low concordance between methods make the benchmarking task difficult. Typically, as exemplified by some of the references cited here, combinations of aligners and variant callers are tested together and with approaches that rely on a well known set of reference variations, like the very well studied human genome (e.g. NA12878 human HapMap genome of National Institute for Standards and Technology (NIST)'s Genome in a Bottle Consortium (ABMMS, 2014; The 1000 Genomes Project Consortium, 2010)). As an addendum, Genome in a Bottle (Zook and Salit, 2011), Genome Comparison and Analytic Testing (GCAT; bioplanet.com (2013)), and Seqbench (Dander et al., 2014) are worthy of praise as invaluable scientific community and crowdsourcing benchmarking efforts which also aim to

improve variant calling accuracy.

Another kind of approach, exemplified by *BAYSIC* (Cantarel et al., 2014), tries to obtain a consensus result by combining outputs of different variant calling programs (e.g. GATK (McKenna et al., 2010; DePristo et al., 2011), SAMtools (Li et al., 2009a), Atlas (Challis et al., 2012), FreeBayes (Garrison and Marth, 2012), etc.), as well as (optionally) known curated results available in typical SNP information databases (e.g. dbSNP (NCBI, 1998)). This kind of solution aims to provide even greater accuracy for variant calls made by algorithms that may already be rather sophisticated.

Additionally, to avoid some of the inherent limitations faced by the mapping-based variant calling approach, other innovative tools (e.g. Cortex (Iqbal et al., 2012), Bubbleparse (Leggett et al., 2013), Platypus (Rimmer et al., 2014), KisSplice pipeline (Maestre et al., 2015), etc.) have been proposed. Such solutions make use of one or more techniques like *de novo* assembly, coloured de Bruijn graph walking, local assembly, haplotype generation, and local realignment to perform the SNP calling task and improve accuracy even if dealing with non-model species which lack a high quality reference genome.

This leads on to the issue of parametrisation and filtering. When introducing their *SNP-o-matic* tool, Manske and Kwiatkowski (2009) pointed out that "the discovery of SNPs and other variants depends on the alignment algorithm allowing

some mismatches to the reference sequence". Indeed, if one allows only perfect matches, no SNPs can be detected (Altmann et al., 2012). Allowing too many mismatches, though, may lead to incorrect alignments (and hence false positive calls), so maximising the number of aligned reads may not be always the best approach (Altmann et al., 2012). This mismatch rate can be referred as the *mapping stringency* and is typically adjusted by the end user via mapper parameters. This setting controls the algorithm's strategy to find inexact matches. Different algorithms use different strategies to find such inexact matches by allowing a certain number of mismatches (Yu et al., 2012; Li et al., 2008b, 2009c; Langmead et al., 2009; Li and Durbin, 2009, 2010). Furthermore, independently of the mapper strategy and accuracy, the mapping stringency is, in itself, a difficult parameter for the end user to gauge. As exemplified by Nielsen et al. (2011), the choice of the optimal number of mismatches may differ greatly between organisms: e.g. populations of *Drosophila melanogaster* vary much more than human populations, hence the mapping criteria should be different. Parameters for analysis of human data would be too stringent for *D. melanogaster*, decreasing the read depth and, consequently, underrepresenting regions harbouring natural polymorphisms. Conversely, using settings adapted for the latter will be too relaxed for the human dataset, leading to a large number of incorrectly mapped reads and, consequently, more false positive events. Altmann et al. (2012) make the same comparison

with *Mus musculus* strains and human samples. Both studies also point out that the issue is true even within the same species. They cite, for instance, the case of the major histocompatibility complex (MHC), which shows high variability between human individuals. Thus, it is reasonable to assume that these different algorithmic strategies and parameter sets will have some degree of influence on the generation of FP SNPs later on in the downstream analysis.

The same is true regarding the read *mappinq quality* factor. The concept was introduced by Li et al. (2008a), along with the release of their MAQ aligner, as a measure of confidence that a read really belongs to the position it aligned to the reference sequence. It is generally estimated by considering various factors, such as the number of base mismatches and the sizes of inserted or deleted regions in the alignment (Ruffalo et al., 2012). More precisely, it is the Phred-scaled probability (Ewing et al., 1998; Ewing and Green, 1998) of the alignment query sequence being placed at a wrong position (Li and Durbin, 2009, 2010). This measurement is determined by the mapping algorithm, aiming to flag potential ambiguities or any suspected lack of accuracy in the alignments. So, as highlighted by Nielsen and co-workers (2011), it is important for mapping algorithms to cope with inherent NGS errors, as well as with potentially true polymorphisms between the reference and the reads, but also to produce well-calibrated mapping quality scores, as further variant calls and associated posterior probabilities computations

depend on them. The issue here, though, is that not all aligners generate mapping qualities (Ruffalo et al., 2012; Yu et al., 2012). For instance, MAQ (Li et al., 2008a), BWA (Li and Durbin, 2009), BWA-SW (Li and Durbin, 2010), Novoalign (Novocraft, 2008), and SSAHA2 (Ning et al., 2001) do output mapping quality values, but SOAP (Li et al., 2008b), SOAP2 (Li et al., 2009c), Bowtie (Langmead et al., 2009), and BLAT (Kent, 2002) do not. Furthermore, those tools that are capable of reporting mapping quality scores typically employ different strategies to compute and/or report it (Yu et al., 2012). For instance, Li and Durbin, when introducing their popular BWA mapper (Li and Durbin, 2009), cite that the algorithm's strategy is similar to that of MAQ (Li et al., 2008a), except for the fact that in BWA it is assumed that the true hit can always be found. Their rationale for this was that MAQ's formula overestimates the probability of missing a true hit and consequently underestimates the mapping quality. In summary, MAQ's mapping quality is underestimated, while BWA's is overestimated. When BWA-SW was released to deal with longer reads, due to specific heuristic rules deployed, a new formula to approximate the mapping quality was implemented (Li and Durbin, 2010). Yu and co-workers (2012) also discuss the different strategies and values used by BWA and Novoalign to compute the mapping quality. For example, when a read is aligned to a unique position with less than 2 mismatches, BWA reports a score value of 37 and scores between 23 and 0 are given when

reads are aligned to multiple locations. For Novoalign, the best alignment receives a score of 150 while aligments to multiple places receive 0.

Apart from the mentioned differences, the work of Ruffalo et al. (2012) expresses concerns about the fact that many genuine mappings are underestimated because mapping quality scores reported by the tools often do not correlate well with actual likelihood accuracy. Because of this, they propose a machine learning tool, LoQuM (LOgistic regression tool for calibrating the QUality of short read Mappings), to assign reliable scores to mappings of Illumina reads returned by typical aligners. The authors claim that the proper recalibration of the mapping quality scores 'ressurect' many of the 0-labelled mappings, thus enhancing the precision of called SNPs. Since the reliability of read alignments can substantially affect the accuracy of the detection of variations (Li et al., 2008a), dealing with the *mapping quality* problem appropriately is likely to have an effect on rates of FP SNPs.

In an alignment, depths of coverage there are much higher (associated with particular regions of the reference sequence) than the average read depth may indicate off-site mapping of reads (e.g. for paralogs and repetitive sequences) (Myles et al., 2010; Krueger and Andrews, 2012). Another cause for genomic regions showing unexpectedly high read depths are PCR (Polymerase Chain Reaction) duplicate read artefacts (Li et al., 2009b). FP SNPs may arise as a consequence of such heavily over-represented locations. Visualisation tools, like

the Integrative Genomics Viewer (IGV; Robinson et al. (2011)) and Tablet (Milne et al., 2010, 2013a), are great resources for spotting such deviations. Furthermore, variant callers typically use some sort of annotation to express the read depth at a variant locus (Broad Institute, 2012b, 2014c). An example of this coverage annotation is the "DP" (read depth at the position) tag, which may be provided in a VCF file (Danecek et al., 2011b). The computed value can then be used to filter sites covered by excessive numbers of reads (Li, 2014b). But, as this same work highlights, citing the Platypus tool (Rimmer et al., 2014) as an example, different callers may define the depth differently. The evaluation of such computational variability and to what extent the filtering of read depth enables the reduction of false positive events is therefore of great interest.

Independently of other factors, the potential effect of read length on the generation of FP SNPs is also an intriguing aspect. Since the inception of NGS technologies and the creation of the first aligners/SNP callers devoted to them, it has been assumed that the short length of NGS readouts can be considerably challenging for obtaining accurate alignments, especially in scenarios of highly polymorphic genome sequences (Manske and Kwiatkowski, 2009). If an alignment is not reliable, nor will be the variant calling process which, in turn, may lead to false positive events. Nielsen et al. (2011) also emphasize that alignments are more difficult to obtain for regions with higher levels of discrepancy between the

reference genome and the sequenced one. They also remark that this difficulty can be minimised by the use of paired-ended and longer reads. Li et al. (2009b) also mention that paired-end sequencing and increased read lengths enable accurate identification of small indels from Illumina sequencing. Longer reads (and paired-end approaches with longer insert sizes) are supposed to improve the 'mappability' — or uniqueness — of a sequence within a reference genome, a concept which has a major influence on the average mapped depth, thus typically being associated with false-negative single-nucleotide variant calls, and generally showing an inverse correlation with genomic repeats and other problematic (referred to as 'dark matter') regions of a genome (Lee and Schatz, 2012; Derrien et al., 2012; Sims et al., 2014). The BWA-SW work cited above, however, warns about the challenges posed by efficient long read mappings and states that long-read alignment has different objectives from short-read alignment. The authors claim that a long read is more susceptible to structural variations and misassemblies in the reference and that, unlike a shorter read, it is less affected by mismatches close to its end. Because of this, local alignment matches are preferred instead of the full-length short-read alignment. Long-read aligners also must be more permissive to alignment gaps because indels occur more frequently in long reads. It is also natural to think that a longer read is a potential reservoir of more sequencing errors. If these are not entirely removed, it is reasonable to assume that the downstream analysis

will be more susceptible to FP SNP occurrences. On another note, though, it is also one of my assumptions that longer reads have better mapping specificity, resulting in reads remaining unmapped if their true origin is unavailable in the reference sequence. Consequently, longer reads are expected to lead to reduced mismapping and fewer FP SNPs. Since this aspect is very important and poses specific challenges for the aligners, it is a common approach to test for different read lengths, via simulated or real datasets, when benchmarking aligners, variant calling pipelines, and NGS-related accessory tools (Langmead et al., 2009; Li et al., 2009b; Li and Durbin, 2010; Peng et al., 2015). Due to these reasons, read length is also evaluated here along with the aforementioned ones.

As stated by Clevenger et al. (2015), SNP calling results show that not all calls are created equally, suggesting that a variety of factors may generate FP SNPs. In the light of this statement and the complexity involved in the variant calling task, the following question was the main motivation for this chapter's investigation:

- How do the factors mentioned above — reference sequence quality, mapping tool, mapping stringency, variant calling tool, read mapping quality filtering, read depth filtering, and read length — interact and affect the FP SNP generation due to read mismapping?

4.2 Methods

4.2.1 Read datasets preparation

The five chromosome sequences of *Arabidopsis thaliana*, available at
The Arabidopsis Information Resource (2011a), served as the template for the generation of the simulated reads for the study. SimSeq read simulator (last update 4.12.2011; St. John (2014)) was used to generate haploid, error-free paired-end and mate-pair reads (the latter created specifically for the assembly stage) from each of the chromosome sequences (see Appendix C, subsection C.1.1, items C.1.1.1 and C.1.1.2). Following the same idea of Chapter 3, this sampling mode would allow the assumption that every SNP encountered in the mappings must be a FP SNP which is due to read mismapping, as there were no other sources of variant alleles. Paired-end reads were produced with 100-fold coverage depth and at lengths of 50, 100, 150, 300, 500, and 1,000 bp (Figure 4.1). Fragment sizes for these were 90, 180, 270, 540, 900, and 1,800 bp, respectively. Mate-pair reads were produced with 50-fold coverage depth, at a length of 150 bp, with a fragment size of 3,000 bp. Full details of the fragment sizes are provided in Appendix C, subsection C.1.1, Table C.4.

4.2.2 Reference genome assembly

To provide the conditions typical of a non-model organism use case, two reference

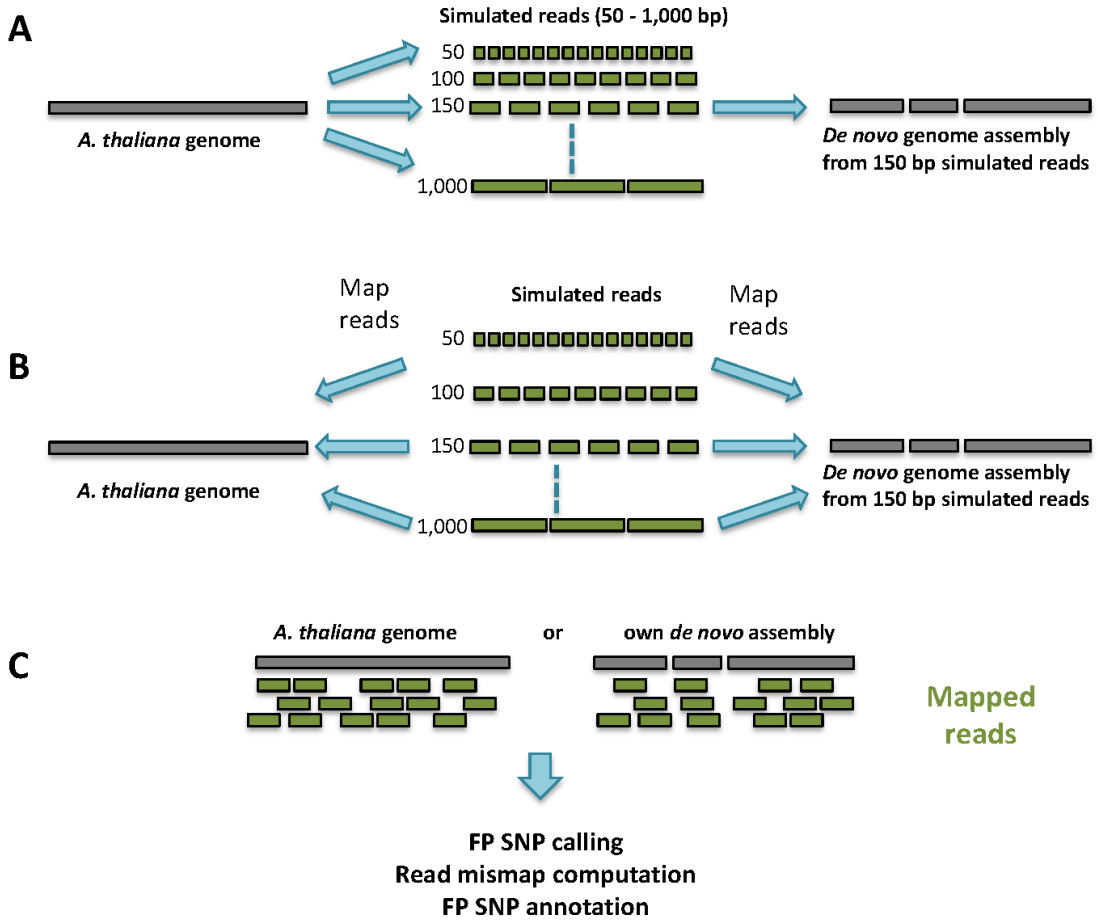

Figure 4.1: Experimental design. (A) The *A. thaliana* genome was used to generate simulated reads of different lengths. *De novo* assemblies were computed from the 150 bp read datasets using different assemblers. (B) With the assemblies as references, separate read mappings were carried out for each of the different read length datasets and with different combinations of factor levels, using the original genome as a control. (C) SNP detection was carried out with different variant callers and the results were analysed to detect whether the mismatched reads causing the SNPs were due to mismapping. SNP annotation was performed to detect enrichment for particular genomic features at SNP positions. Adapted from Ribeiro et al. (2015).

sequences for the read mapping were *de novo* assembled from the 150 bp read datasets, one using the Velvet assembler version 1.2.10 (Zerbino and Birney, 2008) and the other using the Allpaths-LG assembler version r51511 (Gnerre et al., 2011; Ribeiro et al., 2012).

To keep the design of the experiment simple, only the 150 bp read datasets were used for assembly. The depth of coverage for the assemblies was 150x, where 100x was contributed by the 150 bp paired-end reads dataset, while 50x was contributed by the mate-pair reads. Each assembler was run twice, using separately simulated read datasets. Additional information about the assembly process can be found in Appendix C, subsection C.1.2.

QUAST version 2.1 (Gurevich et al., 2013) was used to assess the degree of difference between the *de novo* assembled reference sequences and the *A. thaliana* genome sequence (the control for the read mapping), by analysing each replicate assembly and using the *A. thaliana* genome sequence and the gene models as the benchmark dataset. The assessment results are shown in Appendix C, subsection C.1.2, Table C.5. Definitions of the metrics employed by QUAST are available in the online manual for this software (SPBAU, 2013).

4.2.3 Read mapping

Each of the six read datasets (50–1,000 bp) was mapped to the *de novo* assemblies and the *A. thaliana* control (see below) with Bowtie2 version 2.2.1

(Langmead and Salzberg, 2012) and BWA-SW version 0.7.10-r789 (Li and Durbin, 2010), both very mature and widely used alignment tools (Farrer et al., 2013; Lu et al., 2014), capable of dealing with the range of read lengths explored in the study. Such tools' algorithms are BWT-based and had their differences explored and benchmarked, for instance, in studies like Hatem et al. (2013) and Cornish and Guda (2015).

In order to keep coverage comparable among all mappings, the same mismatch rate was applied across all read lengths, rather than a fixed number of mismatches. To enable any SNPs to be called, at least one mismatch per read needs to be allowed. Taking a read of 50 bp as an example, this means a mismatch rate of 1 mismatch in 50 bp, or 2%. Aiming to compare strict and relaxed mismatch stringencies, the default of the latest BWA algorithm was chosen as the relaxed setting. This was calculated as being equivalent to 14% of mismatches per read. Then, both mismatch rates (2% and 14%) were applied to each of the mappers. Appendix C's subsection C.1.3 describes how the parameter settings were calculated for each mapper.

4.2.4 SNP calling

The FreeBayes variant caller (version v0.9.18-3-gb72a21b; Garrison and Marth (2012); Garrison (2012)) and the Genome Analysis Toolkit (GATK, version 3.3-0 (McKenna et al., 2010; DePristo et al., 2011; Broad Institute, 2012a)) were run over

each of the mappings separately. Both tools were chosen for SNP discovery as they are widely used (You et al., 2012) as general purpose callers and provide substantial configurability. GATK HaplotypeCaller module performs local assembly around variant regions, aiming to improve accuracy. FreeBayes, as explained in Chapter 3, is a Bayesian haplotype-based caller. These methods were also benchmarked, for example, in studies like Cornish and Guda (2015).

To speed up the SNP calling in FreeBayes, a Java SE 7/SAMtools 0.1.18 (Li et al., 2009a) wrapper was produced around it. The wrapper splits and parallelises the job across multiple nodes and processors of a compute cluster. This allowed the jobs to run in a fraction of the time that would otherwise have been required. This is achieved by querying the list of contigs, discarding those that have no reads mapped to them, splitting the remainder into discrete regions that can be processed independently by FreeBayes, before finally concatenating the results back together into a single VCF file.

For GATK, a pipeline script was designed to perform duplicate markup with Picard Tools (version 1.119 (Broad Institute, 2014a)), local realignment around indels, and variant calling with GATK. The base quality recalibration step was left out as there were no known variants as part of the study design. To evaluate the effect of the mapping quality, both variant callers were configured to run with (MAPQ = 20) and without (MAPQ = 0) mapping quality filtering. The detailed

parameters used in FreeBayes and GATK are available in the respective items of Appendix C, subsection C.1.4.

SNPs were also filtered by read depth as an additional experimental factor (maximum read depth 150 *versus* no filtering). As mentioned in this chapter's section 4.1, depth filtering can be applied to remove SNPs located in large accumulations of reads in regions that e.g. represent collapsed repeats in the reference sequence and consequently attract large numbers of reads.

In order to provide more realistic final SNP numbers, multi-allelic SNPs were also removed from all resulting VCF files as well as SNPs with quality scores of less than 20, using a custom bash script. More specifically, if both REF and ALT allele columns of the VCF file had lengths of 1 character (1-based coordinate system), characterising a straight bi-allelic SNP, and column QUAL had a value higher or equal to 20, the corresponding line of the original VCF file was retained.

4.2.5 Control dataset

The same approach as used in Chapter 3 (subsection 3.2.1) was applied here, so the original *A. thaliana* reference sequence would act as a control. As before, the expectation was that the mapping to the original *A. thaliana* reference sequence (the control) would yield fewer FP SNPs than the mapping against the *de novo* assembled reference sequences. Figure 3.6 shows the concept of the control.

4.2.6 Read mismapping quantification stage

The custom Java/Picard API/SAMtools/BLASTN pipeline used in Chapter 3 was refactored to be able to deal with the new read label scheme introduced by SimSeq and, also, to directly process the output from the filtered list of bi-allelic SNPs with quality scores of, at least, 20. Again, the aim was to quantify events where misplaced reads caused SNPs, based on the available read origin information. Thus, for each SNP in each contig, using the same 'read mismapping quantifier' algorithm introduced in Chapter 3, it retrieves the unique covering reads at each SNP site under evaluation and computes the number of reads containing the same allele as the reference sequence as well as the number of reads containing alternate alleles. Based on the SNP site original information (position and allele) retrieved from the BLAST database as well as the read labels' region ranges available from the read simulator, the code determines the percentage of mismapped reads containing the allele alternate to the corresponding allele in the genome, verifies whether the reads originally belong to a different chromosome or different region in the same chromosome, and outputs the results in a tabulated text file. To avoid redundancy, only those SNPs were considered that had not been filtered out by the depth filter. The workflow is shown in Figure 4.2 and the pipeline usage is detailed in Appendix C, subsection C.1.5.

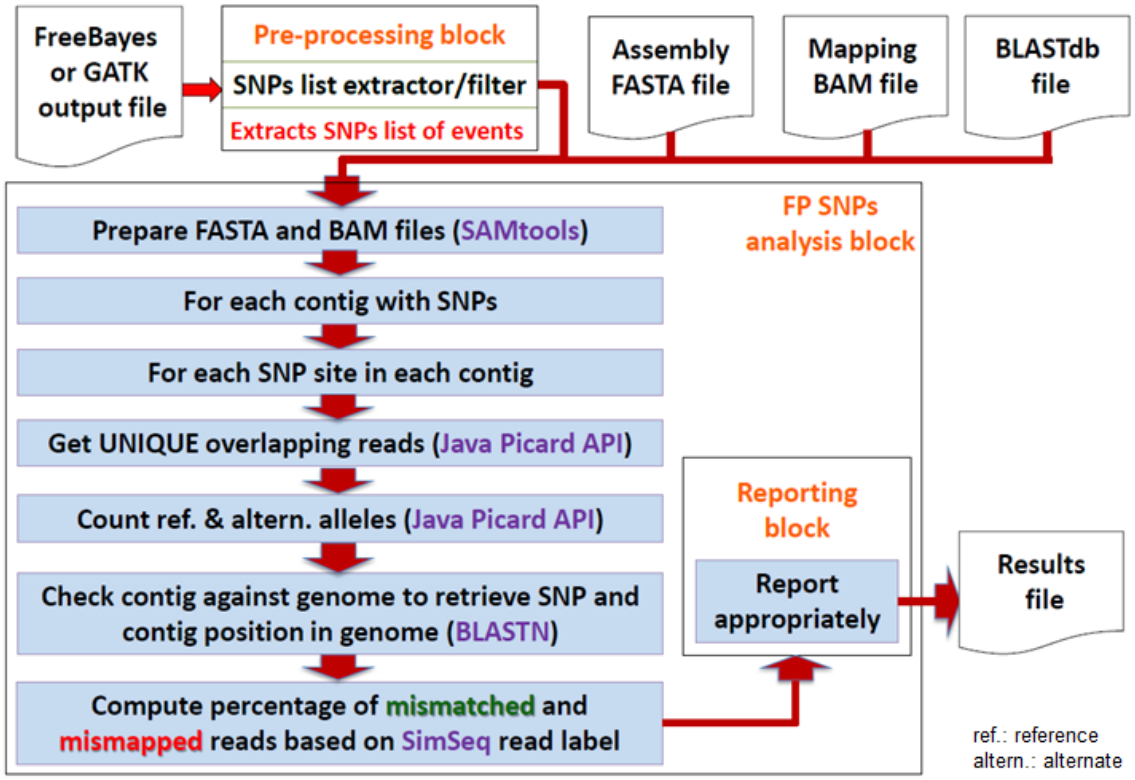

Figure 4.2: Read mismapping quantification code workflow. The program scans for each SNP within a contig and extracts the reads overlapping (covering) each SNP site. It then counts the number of reads containing the same allele as the reference sequence as well as the number of reads containing alternate alleles. It also checks for the original position and allele in the original genome that corresponds to the SNP site in the contig (with the BLASTN parameter -max_hsps_per_subject set to 1). Finally, it computes the percentage of mismapped reads containing the allele alternate to the corresponding allele in the genome, writing the output to a results file. There were cases (approximately 63%, in average, across the assembly replicate runs) where there was disagreement between the allele in the assembled contig and the corresponding position in the original genome sequence (based on the BLASTN analysis), and, in such cases, it was assumed that this was due to a single base misassembly in the contig. In these situations, the reads containing the allele observed in the contig were counted as mismatched, as they contained the allele that differed from the original genome sequence. See Appendix C, subsection C.1.9, for the link to the source code. Adapted from Ribeiro et al. (2015).

4.2.7 SNP annotation

The approach introduced in Chapter 3 was also applied here to test whether

the regions containing SNPs were enriched for particular types of genomic features. The SNP annotation procedure is described in subsection 3.2.1, and used the same BLAST database built in Chapter 2's subsection 2.3.1, item 2.3.1.1. Appendix C, subsection C.1.6 provides more details about how this procedure was carried out.

4.2.8 Replicate workflow runs

To ensure reproducibility and consistency, the experiment was carried out in duplicate. For each read length, two independent, randomly sampled read sets were created, and a new assembly was made from the 150 bp read datasets using both Velvet and Allpaths-LG. The mapping of all read datasets, SNP calling, and the SNP annotation were performed with both the *de novo* assemblies and the whole genome control as reference sequences for each factor combination. Additional information about the replicate assemblies is also available in the Appendix C, subsection C.1.2, Table C.5. Figures 3.6, 4.1, and 4.3 summarise the study's experimental design and the application of tools and variables.

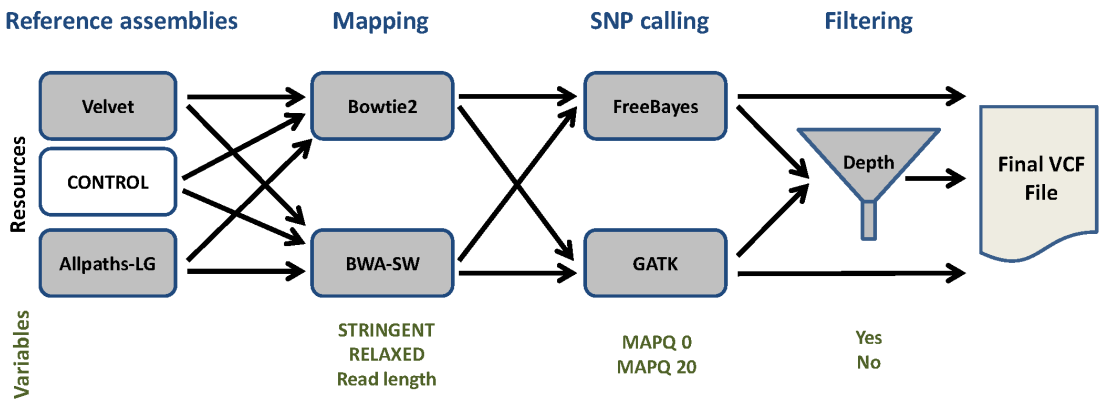

Figure 4.3: Tools and variables used in the experiment. Paired-end datasets of differing read lengths (50–1,000 bp) were mapped using Bowtie2 and BWA-SW with either high (2% mismatches) or low (14% mismatches) mapping stringency. The *de novo* assemblies computed with Velvet and Allpaths-LG were used as references as well as the original *A. thaliana* reference sequence (control). All the resulting mappings underwent SNP calling with the variant callers FreeBayes and GATK, with and without filtering for read mapping quality. The resulting SNPs were filtered by coverage depth (< 150) and these call sets were compared to their unfiltered counterparts. For the final SNP counts, only bi-allelic entries with a SNP quality score greater than 20 were used. Adapted from Ribeiro et al. (2015).

4.2.9 Statistical analysis

Analysis of variance (ANOVA) was used to test for significant effects of the seven factors and all possible interactions on the number of false positives detected. The replicate effect was fitted as a random effect while all other effects and interactions were fitted as fixed effects. The untransformed number of false positives did not satisfy the usual ANOVA assumptions of normally distributed residuals with constant variance. The number of FP SNPs was therefore analysed after a log_{10}(N+1) transformation, which improved the distribution of the residuals. A random permutation test with 999 permutations was also run to obtain a

non-parametric estimate of the significances of each effect, and this gave very similar probabilities to the usual ANOVA F probabilities. The analysis was carried out using GenStat 16 for Windows (Payne et al., 2013).

4.3 Results

4.3.1 General observations

The relationship between FP SNPs and seven factors involved in mapping-based variant calling — quality of the reference sequence, read length, choice of mapper and variant caller, mapping stringency, and filtering of SNPs by read mapping quality and read depth — was explored in the study. This resulted in 576 possible factor level combinations.

The range of FP SNP numbers observed in the experiment varied from 0 to 36,621, depending upon the choice of reference sequence, tools, and parameters. Out of the 576 factor level combinations, 211 contained zero FPs (see file snpNumbersStats.xlsx; Appendix C, subsection C.1.10). These included sets using the BWA mapper on the "strict" mismatch setting with the GATK variant caller for all combinations of depth filtering/no depth filtering, all three assembly types, MAPQ settings of 0 or 20, and the full range of read lengths. Zero FP SNPs were also found for sets using the BWA mapper on the "strict" mismatch setting with the FreeBayes variant caller and a MAPQ setting of 20 for all combinations of

depth filtering/no depth filtering, all three assembly types, and the full range of read lengths. For the control assembly only, the FP count remained at zero in the combinations above even if the "relaxed" mismatch setting was used. The Bowtie2 mapper found zero FPs for the control assembly only and read lengths of 150 bp or fewer, with all combinations of depth filtering/no depth filtering, variant caller, stringency and MAPQ settings, as well as on the "strict" setting with 500 or 1,000 bp reads. None of the mappings against the *de novo* assemblies achieved a zero FP count on the relaxed mismatch setting. At the other end of the spectrum, the largest mean number of FPs encountered was 36,260.5 (300 bp reads, Allpaths assembly, relaxed Bowtie2 mapping, MAPQ filter 0, FreeBayes, no depth filtering).

The majority of factor level combinations in the control group (139 out of 192) contained no FP SNPs at all, and most of the remainder had less than 1,000 FP SNPs (see file snpNumbersStats.xlsx; Appendix C, subsection C.1.10). A large amount of variability was present, however, within the control group, and some call sets contained very large numbers of FP SNPs. The worst performing combination in the control group comprised 300 bp reads mapped with Bowtie2 using relaxed mapping, FreeBayes variant calling, no depth filter, and a MAPQ filter of 0. This yielded an average of 20,471.5 FP SNPs. The equivalent combination of tools, using the strict mapping setting, resulted in an average of only 17 FPs; a reduction of 3 orders of magnitude.

4.3.2 Main effects and interactions among experimental factors

All factors, apart from experimental replicate, had highly significant main effects on FP SNP numbers in the multifactorial ANOVA (Table 4.1 below and file ANOVA_FullResults.xlsx; Appendix C, subsection C.1.10).

Table 4.1: Main effects from the factorial Analysis of Variance (ANOVA). The full list of all possible interaction terms can be found in the file ANOVA_FullResults.xlsx; Appendix C, subsection C.1.10. Adapted from Ribeiro et al. (2015).

Source of variation	d.f.	s.s.	m.s.	v.r.	F prob.	perm. prob.	Percentage SS
replicate stratum	1	0.01693	0.01693	8.82			
replicate.*Units* stratum							
read length	5	40.79358	8.15872	4247.34	0.000	0.001	1.18
assembly	2	1516.31545	758.15772	394688.33	0.000	0.001	43.90
mapper	1	265.90685	265.90685	138428.09	0.000	0.001	7.70
stringency	1	371.94519	371.94519	193630.46	0.000	0.001	10.77
MAPQ	1	55.69223	55.69223	28992.74	0.000	0.001	1.61
variant caller	1	73.45412	73.45412	38239.38	0.000	0.001	2.13
depth filter	1	5.92562	5.92562	3084.81	0.000	0.001	0.17
Residual	575	1.10452	0.00192				

Abbreviations: d.f.: degrees of freedom; s.s.: sum of squares; m.s.: mean square; v.r.: variance ratio; F prob.: F probability; perm. prob.: permutation probability; Percentage SS: Percentage of sum of squares

However, there was a large number of highly significant higher-order interaction terms in the ANOVA results, and these suggested many complex interactions between experimental factors. Figures 4.4 and 4.5 show trellis plots for the two major higher-order interactions that summarise most of the variability attributed to the interaction terms.

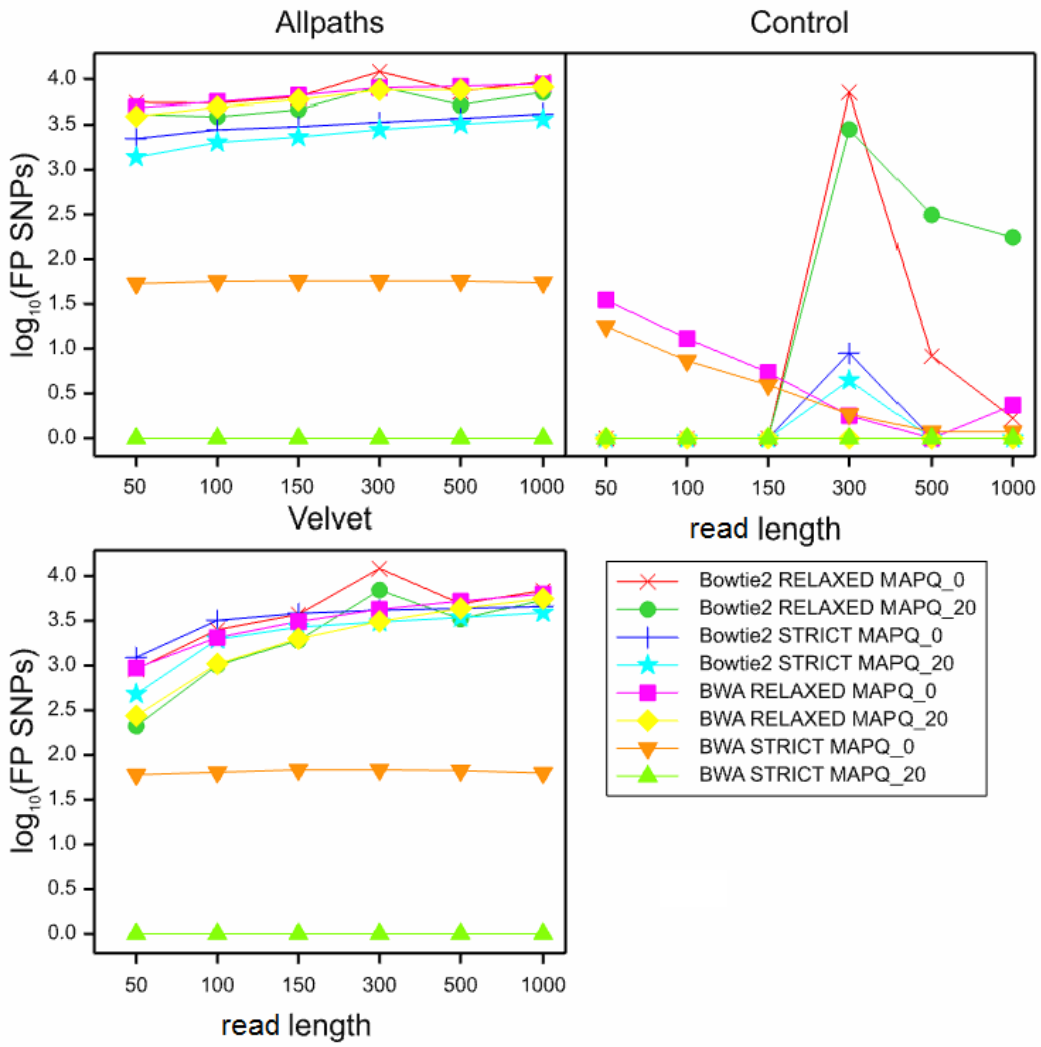

Figure 4.4: 5-way interaction between assembly, mapper, read length, MAPQ, and mapping stringency. Trellis plots for the first major higher-order interaction that summarise most of the variability attributed to interaction terms. Adapted from Ribeiro et al. (2015).

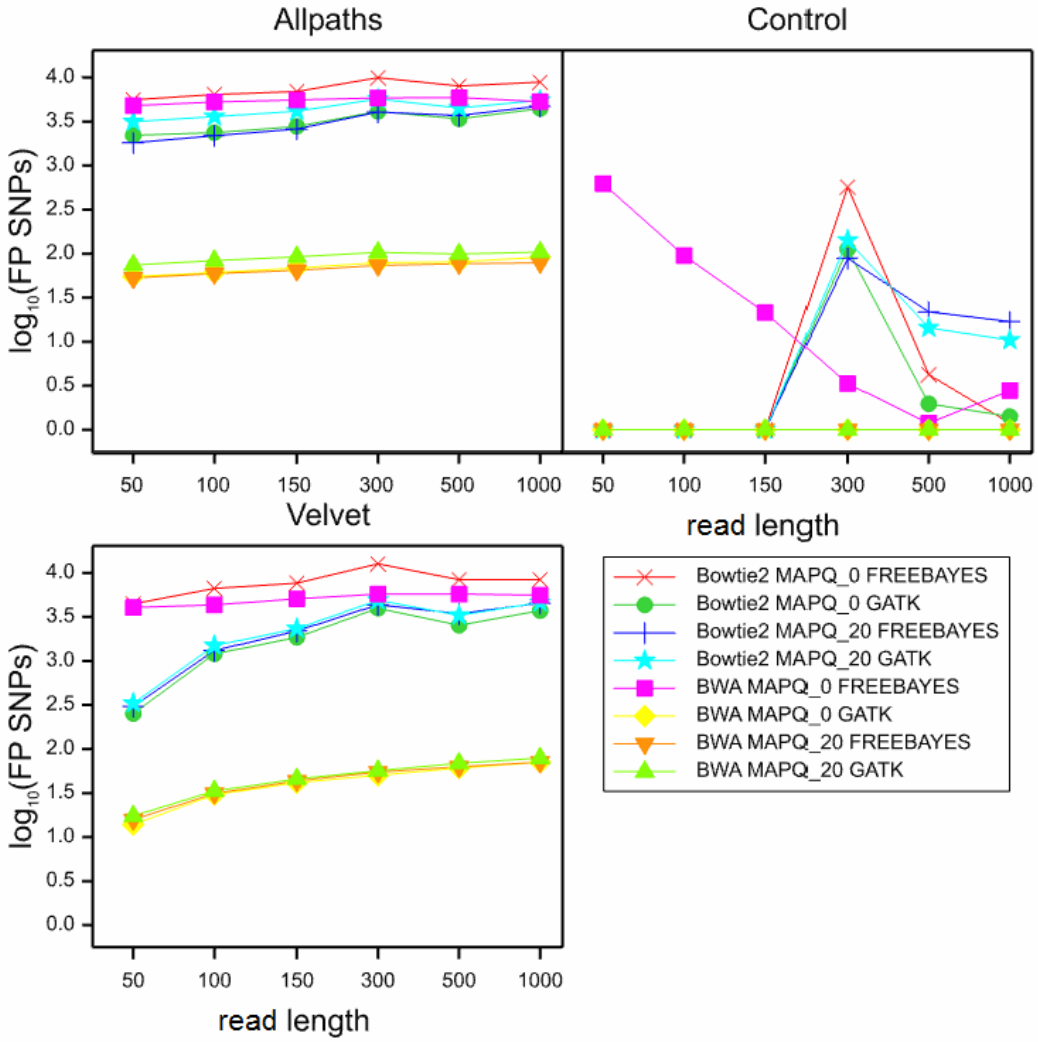

Figure 4.5: 5-way interaction between assembly, mapper, variant caller, MAPQ, and read length. Trellis plots for the second major higher-order interaction that summarise most of the variability attributed to interaction terms. Adapted from Ribeiro et al. (2015).

The equivalent numerical values are shown in Tables 4.2 and 4.3.

Table 4.2: First major higher-order interaction. Log_{10}-transformed means for the 5-way interaction between assembly, mapper, read length, MAPQ, and mapping stringency. Adapted from Ribeiro et al. (2015).

			read length (bp)	50	100	150	300	500	1,000
assembly	mapper	stringency	MAPQ						
Allpaths	Bowtie2	relaxed	0	3.743	3.736	3.807	4.084	3.863	3.975
Allpaths	BWA	relaxed	0	3.692	3.752	3.823	3.906	3.922	3.947
Allpaths	Bowtie2	strict	0	3.349	3.445	3.480	3.529	3.571	3.621
Allpaths	BWA	strict	0	1.729	1.755	1.760	1.759	1.758	1.739
Control	Bowtie2	relaxed	0	0.000	0.000	0.000	3.856	0.920	0.226
Control	BWA	relaxed	0	1.547	1.112	0.736	0.254	0.000	0.369
Control	Bowtie2	strict	0	0.000	0.000	0.000	0.953	0.000	0.000
Control	BWA	strict	0	1.248	0.866	0.595	0.270	0.075	0.075
Velvet	Bowtie2	relaxed	0	2.960	3.399	3.572	4.084	3.691	3.838
Velvet	BWA	relaxed	0	2.972	3.313	3.491	3.628	3.720	3.799
Velvet	Bowtie2	strict	0	3.091	3.504	3.582	3.618	3.638	3.658
Velvet	BWA	strict	0	1.779	1.806	1.834	1.834	1.825	1.798
Allpaths	Bowtie2	relaxed	20	3.615	3.589	3.668	3.917	3.716	3.857
Allpaths	BWA	relaxed	20	3.594	3.695	3.775	3.880	3.882	3.914
Allpaths	Bowtie2	strict	20	3.143	3.306	3.366	3.448	3.507	3.562
Allpaths	BWA	strict	20	0.000	0.000	0.000	0.000	0.000	0.000
Control	Bowtie2	relaxed	20	0.000	0.000	0.000	3.451	2.497	2.246
Control	BWA	relaxed	20	0.000	0.000	0.000	0.000	0.000	0.000
Control	Bowtie2	strict	20	0.000	0.000	0.000	0.648	0.000	0.000
Control	BWA	strict	20	0.000	0.000	0.000	0.000	0.000	0.000
Velvet	Bowtie2	relaxed	20	2.322	3.002	3.281	3.842	3.519	3.733
Velvet	BWA	relaxed	20	2.438	3.018	3.300	3.492	3.637	3.748
Velvet	Bowtie2	strict	20	2.682	3.292	3.429	3.486	3.536	3.590
Velvet	BWA	strict	20	0.000	0.000	0.000	0.000	0.000	0.000

Sed = 0.02191

Abbreviations: Sed: standard error of the difference; bp: base pairs

Table 4.3: Second major higher-order interaction. Log_{10}-transformed means for the 5-way interaction between assembly, mapper, variant caller, MAPQ, and read length. Adapted from Ribeiro et al. (2015).

			read length (bp)	50	100	150	300	500	1,000
assembly	mapper	variant caller	MAPQ						
Allpaths	Bowtie2	FreeBayes	0	3.75	3.81	3.84	4.00	3.90	3.95
Allpaths	BWA	FreeBayes	0	3.68	3.72	3.75	3.77	3.77	3.73
Allpaths	Bowtie2	GATK	0	3.34	3.37	3.45	3.61	3.53	3.65
Allpaths	BWA	GATK	0	1.74	1.78	1.84	1.90	1.91	1.96
Control	Bowtie2	FreeBayes	0	0.00	0.00	0.00	2.76	0.63	0.08
Control	BWA	FreeBayes	0	2.80	1.98	1.33	0.52	0.08	0.44
Control	Bowtie2	GATK	0	0.00	0.00	0.00	2.05	0.29	0.15
Control	BWA	GATK	0	0.00	0.00	0.00	0.00	0.00	0.00
Velvet	Bowtie2	FreeBayes	0	3.65	3.82	3.88	4.10	3.92	3.92
Velvet	BWA	FreeBayes	0	3.61	3.64	3.71	3.76	3.76	3.75
Velvet	Bowtie2	GATK	0	2.40	3.08	3.27	3.60	3.41	3.57
Velvet	BWA	GATK	0	1.14	1.48	1.62	1.70	1.78	1.85
Allpaths	Bowtie2	FreeBayes	20	3.26	3.34	3.42	3.61	3.57	3.68
Allpaths	BWA	FreeBayes	20	1.72	1.77	1.81	1.87	1.89	1.90
Allpaths	Bowtie2	GATK	20	3.50	3.56	3.62	3.76	3.65	3.74
Allpaths	BWA	GATK	20	1.87	1.92	1.96	2.01	2.00	2.02
Control	Bowtie2	FreeBayes	20	0.00	0.00	0.00	1.95	1.34	1.23
Control	BWA	FreeBayes	20	0.00	0.00	0.00	0.00	0.00	0.00
Control	Bowtie2	GATK	20	0.00	0.00	0.00	2.15	1.16	1.02
Control	BWA	GATK	20	0.00	0.00	0.00	0.00	0.00	0.00
Velvet	Bowtie2	FreeBayes	20	2.48	3.12	3.34	3.64	3.54	3.66
Velvet	BWA	FreeBayes	20	1.20	1.50	1.64	1.74	1.80	1.85
Velvet	Bowtie2	GATK	20	2.52	3.17	3.37	3.69	3.52	3.67
Velvet	BWA	GATK	20	1.24	1.52	1.66	1.75	1.84	1.90
Sed = 0.02191									

Abbreviations: Sed: standard error of the difference; bp: base pairs

4.3.2.1 Assembly

The reference sequence used had the most pronounced effect on the rate of FP SNPs, accounting for 43.9% of the total variation in the data (Table 4.1), with

a highly significant main effect. There were significant interactions with all six of the other factors. Mappings against the original *A. thaliana* genome (control) yielded comparatively few FP SNPs in most cases (Figures 4.4 and 4.5), while mappings against the *de novo* assemblies generally produced FP SNP numbers orders of magnitude greater. The Velvet reference sequence slightly outperformed the Allpaths sequence in most cases.

4.3.2.2 Stringency

Mapping stringency accounted for 10.8% of the total variation in the data, making it the second most important factor in the experiment (Table 4.1). The main effect in the ANOVA was statistically highly significant, with the global means suggesting a reduction of approximately one order of magnitude in FP numbers for the "strict" setting (log_{10}-transformed means: relaxed 2.64; strict 1.50). This effect was observable in the majority of interactions analysed here (Tables 4.2, 4.4, 4.6 and Figure 4.4).

Table 4.4: Mapper and mapping stringency interaction. Log_{10}-transformed means for the interaction between mapper and mapping stringency. Adapted from Ribeiro et al. (2015).

	stringency	relaxed	strict
mapper			
Bowtie2		2.7780	2.3343
BWA-SW		2.5099	0.6807
Sed = 0.00365			

Abbreviation: Sed: standard error of the difference

Table 4.5: MAPQ and variant caller interaction. Log_{10}-transformed means for the interaction between MAPQ filter level and variant caller. Adapted from Ribeiro et al. (2015).

	variant caller	FreeBayes	GATK
MAPQ			
0		2.8277	1.7635
20		1.8287	1.8830
Sed = 0.00365			

Abbreviation: Sed: standard error of the difference

Table 4.6: Assembly type, mapper, and mapping stringency interaction. Log_{10}-transformed means for the interaction between assembly type, mapper, and mapping stringency. Adapted from Ribeiro et al. (2015).

	mapper	Bowtie2		BWA-SW	
	stringency	relaxed	strict	relaxed	strict
assembly					
Allpaths-LG		3.7976	3.4439	3.8151	0.8750
Control		1.0996	0.1334	0.3349	0.2608
Velvet		3.4369	3.4256	3.3796	0.9063
Sed = 0.00633					

Abbreviation: Sed: standard error of the difference

The reduction in FP numbers from applying the strict mismatch setting was greatest for the combination of BWA and the two poorer reference sequences, and for the combination of Bowtie2 and the control reference sequence with read lengths of 300–1,000 bp.

4.3.2.3 Mapping tools

This was the third most important factor in FP SNP generation, in terms of the contribution to the overall variation in the data, contributing 7.7% of the

total (Table 4.1). On average, BWA produced fewer FPs than Bowtie2 (log_{10} transformed means: 1.59 *vs* 2.55, respectively), but deviations from this pattern occurred depending on the read length, MAPQ, mapping stringency, and reference sequence (Tables 4.2, 4.3, 4.4 and 4.6; Figures 4.4 and 4.5). Most of these occurred in the relaxed mappings with MAPQ_0 filtering. For the short read mappings (50–150 bp) against the control reference with MAPQ_20 filtering, both mappers performed equally well. However, even on the most conservative settings (strict mapping, MAPQ_20) and with the best reference sequence (control), Bowtie2 performed poorly on the 300 bp reads, whereas on the longer reads (500/1,000 bp) its performance matched that of BWA (Table 4.2).

4.3.2.4 Variant caller

The effect of the variant calling software, again, was statistically highly significant, but had interdependencies with other factors. Global means suggested that GATK produced fewer FPs than FreeBayes but this only held true for the MAPQ_0 call sets. When a MAPQ filter of 20 was applied, the GATK FP rates, in most cases, were either equal to or slightly higher than those obtained with FreeBayes (Tables 4.3 and 4.5).

4.3.2.5 MAPQ-based filtering of SNPs

Read mapping quality based filtering of SNPs (0 *versus* 20) also had a significant main effect, and, while the global means suggested that MAPQ filtering of SNPs

reduces FP numbers (log_{10} means: MAPQ_0 = 2.29; MAPQ_20 = 1.85), this did not apply universally. When filtering for MAPQ_20, FP numbers were reduced for the FreeBayes call sets but not for GATK call sets (Table 4.5).

4.3.2.6 Read length

FP SNP numbers did not strictly decrease as a function of read length (Figures 4.4 and 4.5). Actually, FP SNP numbers, in most call sets, were either flat when plotted against read length, or showed an asymptotic increase with read length. Only the BWA/MAPQ_0 call sets in the control group showed a decline of FP numbers with read length, with a minimum at 500 bp and a slight increase at 1,000 bp. In the Control group only, the Bowtie2 mappings had a sharp peak in FP numbers for read length 300 bp, with the 500 bp and 1,000 bp FP numbers still higher than those for the shorter reads (50–150 bp), all of which had zero FPs regardless of any other factors.

4.3.2.7 Depth filter

Filtering SNPs for read depth greater than 150x coverage resulted in lower FP numbers, and the main effect for this was statistically highly significant (Table 4.1). The magnitude of this effect depended on the quality of the reference though, as shown in Table 4.7. The effect of applying depth filtering was strong for the two *de novo* assemblies but relatively small for the control mappings against the intact *A. thaliana* genome.

Table 4.7: Assembly and depth filter interaction. Log_{10}-transformed means for the interaction between assembly and depth filter. Adapted from Ribeiro et al. (2015).

depth filter	no	yes
assembly		
Allpaths	3.1273	2.8385
Control	0.4634	0.4509
Velvet	2.8516	2.7226
Sed = 0.00447		

Abbreviation: Sed: standard error of the difference

4.3.3 Read mismapping statistics, SNP annotation, and genomic distribution of FP SNP sites

The proportion of mismapped reads among reads with alternate alleles at SNP locations was approximately 89% when averaged across all mappings containing FP SNPs (see file avgPctOfMismapping.xlsx; Appendix C, subsection C.1.10). Regions associated with FP SNPs were significantly enriched for transposable element sequences (approximately 30%) (Figure 4.6 below and Appendix C, subsection C.1.7), compared to approximately 6% in the whole genome annotation.

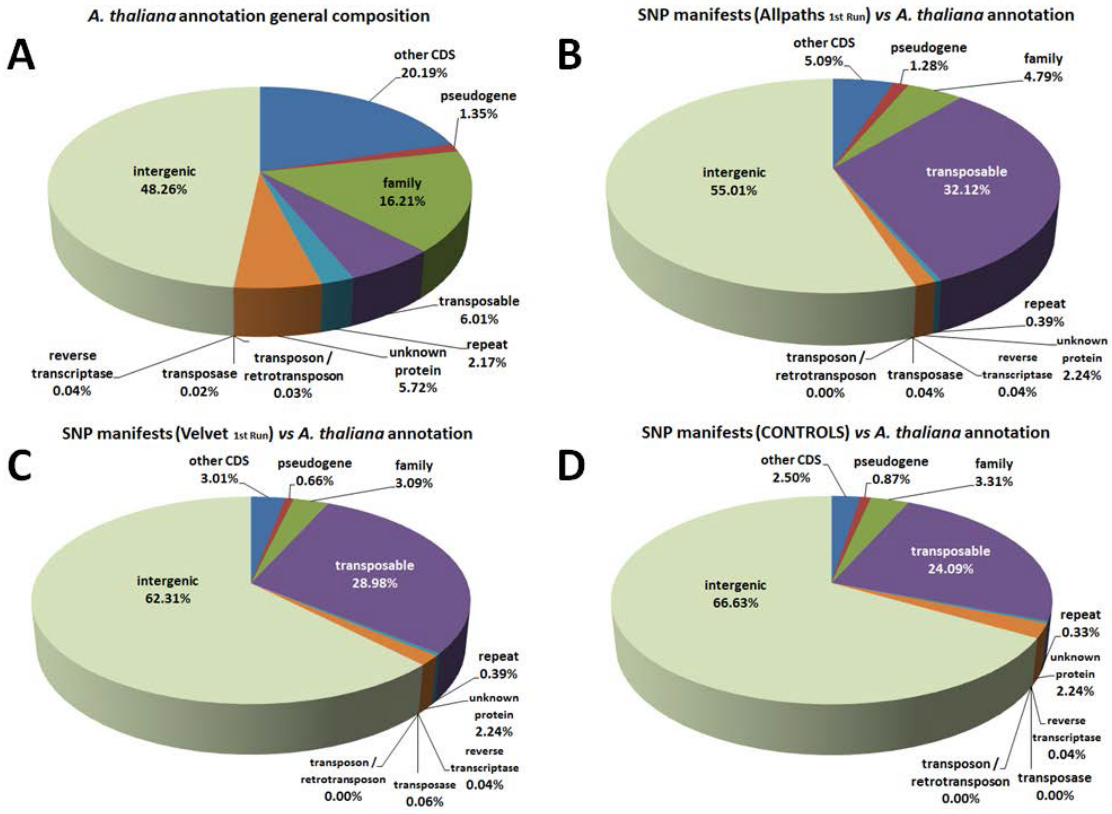

Figure 4.6: SNP annotation. (A) General composition of the *Arabidopsis thaliana* annotation compared with the BLAST-based annotation results for the SNP manifests from the first run replicates of (B) Allpaths-LG, (C) Velvet, and (D) the control runs (compiled). Adapted from Ribeiro et al. (2015).

The distributions of the FP SNPs on the five *A. thaliana* chromosomes are shown in Figure 4.7.

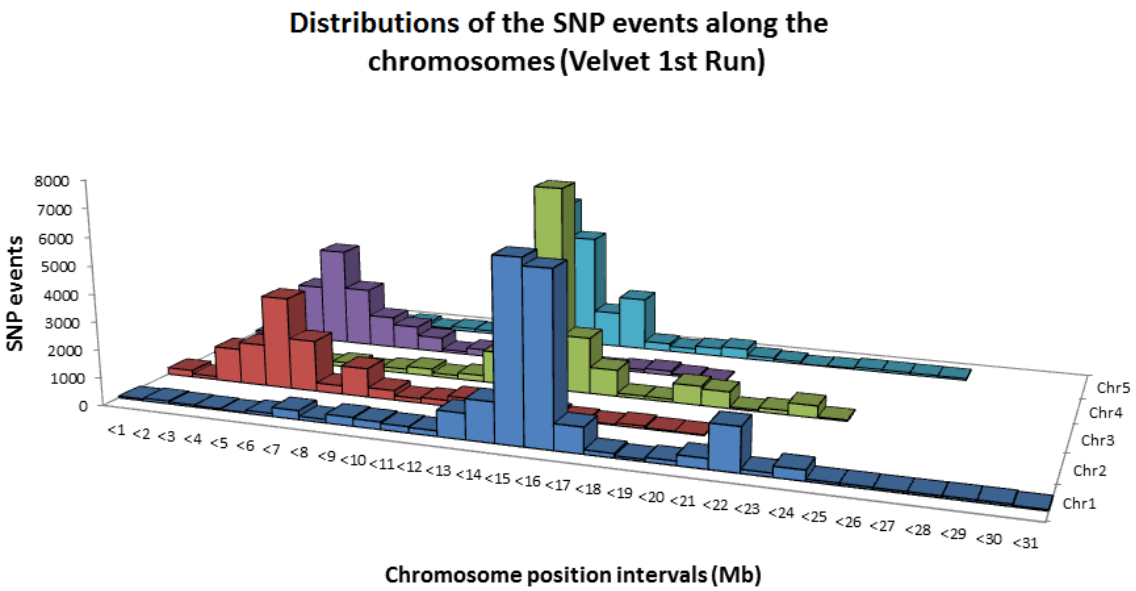

Figure 4.7: FP SNP sites genomic locations (first Velvet *de novo* assembly replicate). Plot of the distributions of FP SNP sites, by chromosome, from the mapping to the first Velvet *de novo* assembly replicate (see Appendix C, subsection C.1.8, for data from other runs). Genomic locations are shown on the x axis divided in intervals of up to 1 mega base pairs (only upper limits depicted for simplicity). FP SNP counts are shown on the y axis.

The great majority of FP SNPs were found in the central (pericentromeric) regions of chromosomes and there was a strong enrichment for transposable element-derived sequences associated to the FPs — approximately 30% for the mappings to the *de novo* assembled reference sequences.

4.4 Discussion

Employing the strategy shown in Figure 4.1, sets of simulated reads of varying sizes were sampled from the *A. thaliana* genome sequence. These reads were error- and variant-free to ensure that every SNP found was indeed a false positive. To

investigate the effect of assembly on FP SNP generation, two different reference sequences were generated using the *de novo* assemblers Velvet and Allpaths-LG. To investigate variations in read mapping, the simulated read sets above mentioned were then mapped to the two *de novo* genome assemblies, as well as the *A. thaliana* reference genome, using two widely used read mappers, Bowtie2 and BWA. The range of read lengths chosen covers most of the currently available sequencing technologies, with the exception of Pacific BioSciences and Oxford Nanopore (Glenn (2011) and updates at The Molecular Ecologist (2014)). The latter two technologies produce longer reads but are currently associated with substantial error rates and their use in variant calling is still in its early stages. The mappings generated were then processed with two popular variant callers, GATK and FreeBayes.

All the tools used here, in each stage of the NGS SNP calling workflow, are remarkable examples of consolidated and robust methods, all were benchmarked in different studies, provide substantial configurability, and have good documentation and support. Therefore, they were selected as being good representatives of the typical methods used in the NGS variant calling scenario.

As a summary of the experiment's results, the variation in the number of FP SNPs generated ranged from 0 to approximately 36,621 for the ∼120 million base pairs (Mbp) genome. Using a fragmented reference sequence led to a huge

increase in the number of FP SNPs generated, as did relaxed read mapping and a lack of SNP filtering. The choice of reference assembler, mapper, and variant caller also significantly affected the outcome. The effect of read length was more complex and suggests a possible interaction between mapping specificity and the potential for contributing more false positives as read length increases. All of the experimental factors tested had statistically significant effects on the number of FP SNPs generated and there was a considerable amount of interaction between the different factors. Nevertheless, since global means hide much of the complexity of the findings, the results of the study should be interpreted in the context of these interactions which are explored in more detail here.

4.4.1 Role of the reference sequence in the generation of FP SNPs

As seen in the pilot study described in Chapter 3, misassembly/non-assembly of the reference sequence facilitate the read cross-mapping and, consequently, the generation of FP SNPs. In that case, the FP SNP numbers were 52-fold lower in the mapping against the original genome in comparison to the mapping against the genome assembly. Therefore, the role of the reference sequence in read mismapping and FP SNP generation was one of the main factors to be explored here.

The difference in FP SNP numbers with the *de novo* assembled reference sequences amounted to several thousands as a result of misassembly or non-assembly

alone. As highlighted in the previous chapter, most existing model organism reference sequences are not comparable to those of non-model organisms in terms of completeness and correctness. The latter are often based on short read technology only, and are subject to little, if any, curation after the initial assembly. When subsequently used as reference sequences for mapping and SNP discovery, these genomes contain numerous candidate regions for mismapping which may induce FP SNPs that look insconspicuous in every respect and are hence difficult to remove by filtering. It is important to notice that the genome used here is small (approximately 125 Mbp) and contains relatively few repeats (The Arabidopsis Genome Initiative, 2000). The effects observed here are likely to be much more pronounced with larger, more complex genomes where misassembly is much more prevalent. Large, complex genomes of this kind are common in plants (Hamilton and Buell, 2012) and other organisms.

There were also significant numbers of FP SNPs in some of the control call sets, based on mapping against the *A. thaliana* sequence. This was surprising, but seemed to be mostly due to certain unfavourable combinations of tools and parameters. The majority of call sets in the controls (282 out of 384) contained no FP SNPs at all, and most of the remainder had less than 1,000 FP SNPs. All of the control call sets with more than 1,000 FP SNPs (n = 20) were run with the relaxed mapping settings, which emphasizes the importance of conservative

mapping even when the reference sequence is well assembled.

4.4.2 Choice of tools for assembly, mapping, and variant calling and their influence on the generation of FP SNPs

This study did not aim to compare the performance of specific tools involved in variant calling, but rather to provide proof of principle that false discovery rates in SNP calling can be significantly affected by the quality of reference sequence, tool choice and parameters. Equally, the current study did not aim to explore whether longer reads, or indeed longer read fragments, provide better *de novo* assemblies, as this has been covered elsewhere (Chaisson et al., 2009; Chang et al., 2014).

The assembly tools used for producing the *de novo* reference sequences comprised Velvet and Allpaths-LG. Velvet is one of the first generation of short read assemblers but has had continuous improvements and updates over many years (Zerbino and Birney, 2008; Zerbino et al., 2009). Allpaths-LG is a relatively recent tool and developers have taken a new approach by requiring input of at least two different fragment size libraries to ensure a high quality assembly. Allpaths consistently performed well in both of the Assemblathon competitions (Earl et al., 2011; Bradnam et al., 2013), so it was surprising that the reference sequence produced by this tool was inferior to that produced by Velvet for most of the major metrics in the QUAST analysis (N50, assembly length, # misassemblies, genome fraction, # genes, largest contig), and that it consistently yielded greater numbers of FP

SNPs than the corresponding Velvet assemblies. It would be interesting as part of future work to explore the reasons behind the difference in performance of the two assemblers.

The two mapping tools used here, Bowtie2 and BWA, are arguably among the most commonly used tools for short read mapping (Farrer et al., 2013; Lu et al., 2014). Both provide a good trade-off between accuracy and performance (Fonseca et al., 2012; Otto et al., 2014) and are mature tools. On average, BWA performed better in this study, but when mapping short (50–150 bp) reads against the good quality control reference sequence with MAPQ_20 filtering, both tools performed equally well, giving zero false positives.

Overall, the variant calling tools performed in similar ways. GATK performed better than FreeBayes with the MAPQ_0 call sets but slightly worst with the MAPQ_20 filtering. The interdependency of the variant caller factor with the others suggest that it may be useful to test different combinations of tools and parameters, eventually in some sort of sub-sampled dataset, in order to assess their overall behaviour when dealing with a particular research question.

Regarding the worst performing factor combination for the control reference sequence (300 bp reads, Bowtie2, relaxed mapping setting, FreeBayes, without depth filtering, and MAPQ = 0), it is worth mentioning the 3 orders of magnitude difference in FP SNP numbers (20,471.5 against 17) obtained by simply changing

the mapping stringency factor to the strict setting. This is a powerful illustration of the drastic effect of mapping stringency on FP SNP discovery.

4.4.3 The impact of SNP filtering on FP SNP numbers

Filtering by MAPQ and maximum read depth both cut FP SNP numbers significantly. Their contribution to the overall variation in the data was relatively small but it is very clear from the data that these filters should be applied wherever it is appropriate. The effect of MAPQ filtering was less explicit — applying the MAPQ_20 filter to the GATK callsets actually increased FP numbers slightly in this experiment. This is counterintuitive and requires further investigation. For the FreeBayes call sets, FP numbers did drop when the MAPQ_20 filter was applied, so it is clear from these results that the filter should be applied when using this variant caller.

4.4.4 The impact of read length on FP SNP numbers

The numbers of FP SNPs observed as a function of read length contradicted the initial assumption that longer reads lead to fewer FP SNPs due to higher mapping specificity and therefore reduced mismapping rates. This was only true for the two MAPQ_0 BWA mappings against the control reference sequence. For most of the other call sets, FP SNP numbers increased with read length. In the Bowtie2 mappings against the control reference sequence, the pattern observed had a sharp peak for the 300 bp read mappings. The potential to cause FP SNPs seems to be

related to the length of the read, providing that reads are mapped with the same mismatch *rate* as length increases, as it was the case of the experiment designed here. Every mismatch with the reference has the potential to become a FP SNP, if suitable numbers of reads are mismapped together, and, in theory, both longer reads and greater mismatch rates contribute to make the problem worse (Figure 4.8).

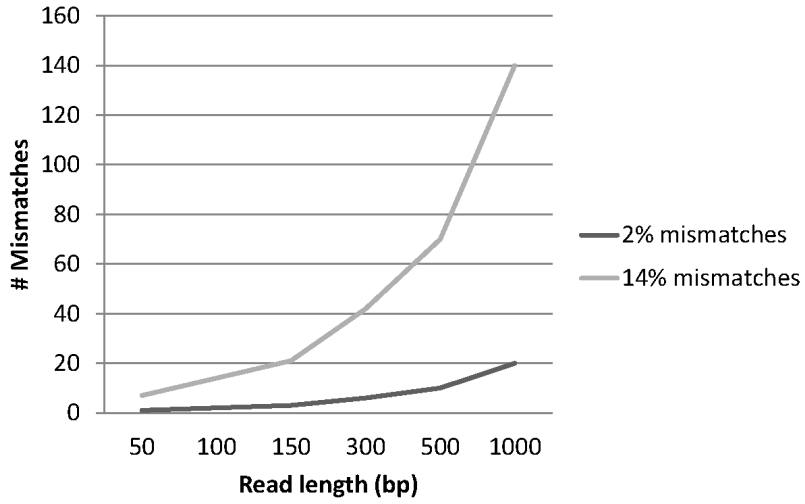

Figure 4.8: Mismatches *versus* read length. Numbers of theoretically possible mismatches per read as a function of read length and mismatch settings. Abbreviations: #: Number of; bp: base pairs. Adapted from Ribeiro et al. (2015).

This is also illustrated by the example shown in Figure 4.9 below.

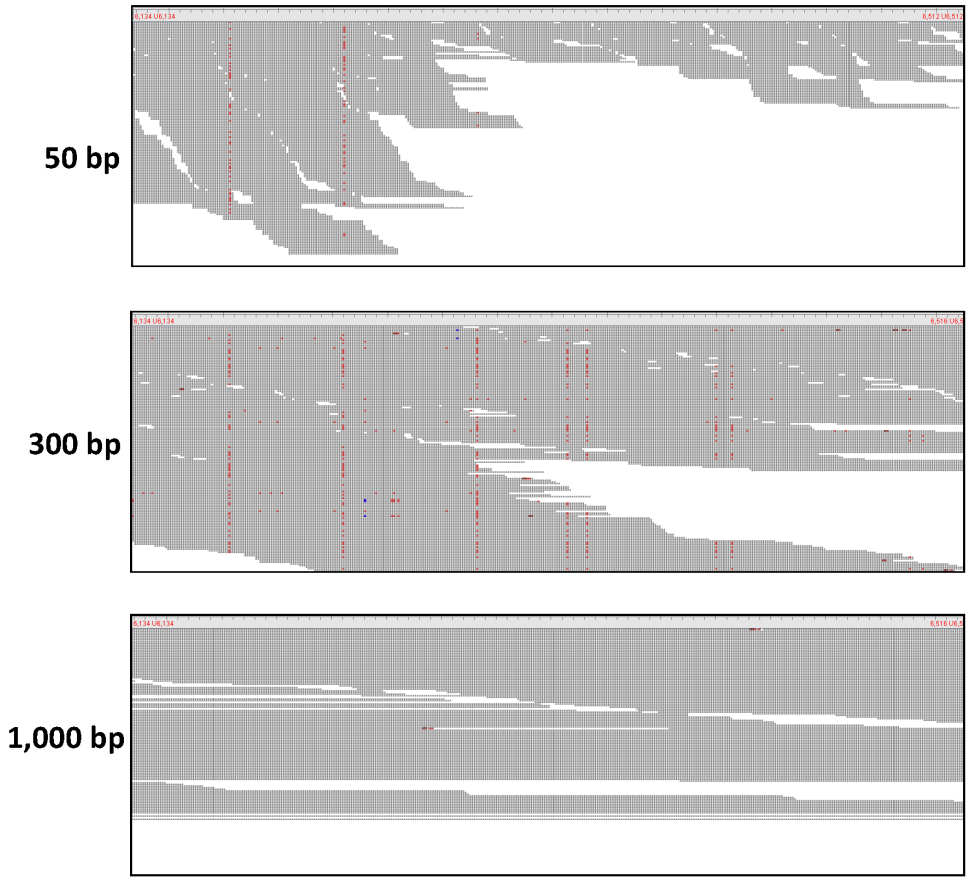

Figure 4.9: Tablet screenshots of read mismapping and corresponding FP SNPs. All screenshots show the same region on chromosome 1, which has been mapped with reads from the correct region on chromosome 1, but also reads from chromosome 2. FP SNPs are visible as vertical, red dotted lines. In this example, the 50 bp reads (top) introduce a small number of FP SNPs, the 300 bp (middle) reads introduce a substantially larger number, but in the mapping of the 1,000 bp reads (bottom) there are no FP SNPs, presumably indicating that the 1,000 bp reads from the contaminating region on chromosome 2 contain too many mismatches to be mapped here. Abbreviation: bp: base pairs. Adapted from Ribeiro et al. (2015).

Here, screenshots from Tablet (Milne et al., 2010, 2013a) show the same region in mappings of different read lengths (only 50, 300, and 1,000 bp shown for brevity) for a given same factor level combination. This is a region that is clearly prone

to read mismapping and, if one considers only the 50 and 300 bp mappings alone, the assumption would be that the longer the reads, the more FP SNPs would be generated. However, the 1,000 bp read mapping shows no signs of SNPs, and it appears that the counterpart 1,000 bp reads from the region that contributes the cross-mapped reads in the 50 and 300 bp mappings simply have too many mismatches to be mapped here. This suggests that greater mapping specificity does play a role in the example, and, for this particular region, the use of longer reads has prevented mismapping and the consequent FP SNPs. Visualisation of the data has produced many other examples where the 1,000 bp mapping instead contained even larger numbers of FP SNPs than any of the comparable shorter read mappings. There were also cases where the 50 bp mapping was the only one containing any FP SNPs at all. Taking both scenarios into consideration, this indicates that the underlying sequence context influences the potential for longer reads having greater mapping specificity and, consequently, whether or not read length makes a difference.

With that said, the potential of the longer reads to cause greater damage seems to be alleviated, at least to some extent, by their greater mapping specificity — the rate of increase of FP SNP numbers with read length in the experiment (Figures 4.4 and 4.5) was not as pronounced as could be expected from what is theoretically possible (Figure 4.8). As already mentioned, the original assumption was that

longer reads map more specifically, thereby reducing the potential for mismapping. The expectation would then be that longer reads have lower rates of mismapping than shorter reads. Due to the availability of the read origin information, it is easy to track the corresponding information about mismapping. By analysing the rates of mismapping for each call set, these can be plotted as a function of both read length and assembly (Figure 4.10).

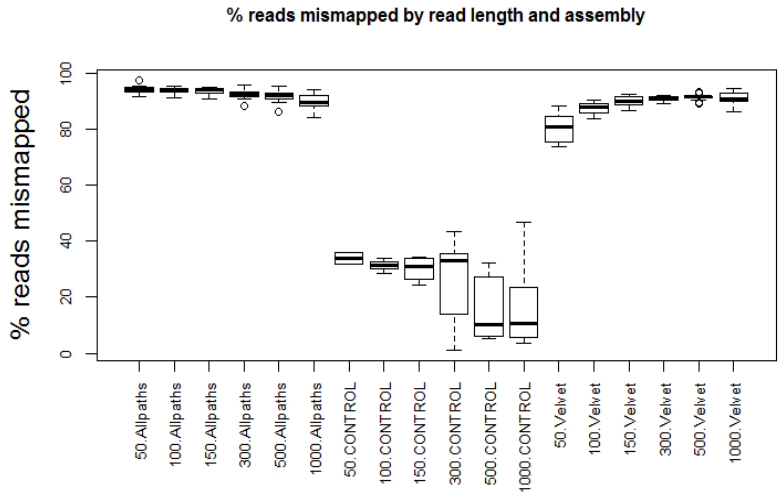

Figure 4.10: Percentages of mismapped reads as a function of read length and type of reference assembly. Boxplots show means (thick black horizontal bar), 25^{th} and 75^{th} centiles (ends of rectangles), 10^{th} and 90^{th} centiles (whiskers) plus individual outliers (circles). Adapted from Ribeiro et al. (2015).

Interestingly, the relationship between read length and rates of mismapping appeared to depend on the reference sequence used. For the Allpaths-assembled reference and the controls, rates of mismapping appeared to decline with increasing read length (Figure 4.10). For the Velvet-assembled reference sequences, this trend

appeared to be reversed and the underlying mechanism for this still requires further investigation.

Overall, the complexity revealed in the read length scenario suggests that there are probably two opposing forces involved here. On the one hand, there is the potential for longer reads to cause greater number of FP SNPs by introducing greater numbers of mismatches. On the other hand, greater mapping specificity in longer reads may mean fewer reads getting mismapped as read length increases, with an accompanying decrease in the likelihood of SNPs being called due to low alternate allele numbers. Within the current experiment, there were no simulated reads of the kind of lengths that are now being generated by e.g. the Pacific Biosciences (Pacific Biosciences of California, Inc., 2015) and Oxford Nanopore (Oxford Nanopore Technologies, 2008) technologies, and it would be highly interesting to explore, in future experiments, whether mapping reads of several kilobases in length genuinely improves mismapping.

4.4.5 Genomic patterns of FP SNP locations

Corroborating what has been seen in the conceptual study of Chapter 3, regions containing FP SNPs were strongly enriched for transposable elements and a large proportion of FP SNPs was located in the pericentromeric regions of the chromosomes, where such repetitive sequences are common (The Arabidopsis Genome Initiative, 2000). This suggests that FP SNP generation is predominantly

associated with the inability of genome assemblers and read mappers to deal with highly repetitive genome sequences and that misassembly and/or non-assembly of repeats or members of gene families in *de novo* genome assembly were the prime causes of FP SNPs in the study.

4.4.6 Taking false negative SNPs into consideration

It is plausible to assume that the combination of tools and parameters used to control FP SNPs occurrences will also influence the levels of false negative (FN) SNPs obtained. As seen in the introductory part of this chapter, based on the work of Cantarel et al. (2014), it is challenging to establish a standard protocol for assuring the highest specificity necessary to minimise false positive calls as well as the highest sensitivity to avoid the false negative ones.

In this study, there were 211 factor combinations for which the FP SNP level was zero. Assuming these as the potential 'best' performing combinations, another experiment was carried out aiming to shed some light on the impact regarding FN SNPs generation. For this, only the available zero-FP SNP combinations related to the 'control' reference genome (the more accurate reference sequence and hence with fewer candidate regions for read mismapping and consequent FP SNPs) were firstly taken into consideration. Then, in order to cover the broadest range of available combinations in terms of read length, mapping, and variant calling tools, 13 "most stringent/most relaxed" pairs of those were selected, resulting in 26 test

combinations for the new experiment (Table 4.8).

Table 4.8: Zero-FP SNP combinations related to the control genome which were selected for the FN SNP experiment categorised by stringency scenario.

Zero-FP SNP CONTROL combinations selected for the FN SNP experiment	
Most stringent scenario	Most relaxed scenario
1,000-CONTROL-BWA-STRICT-20-GATK-yes	1,000-CONTROL-BWA-RELAXED-0-GATK-no
500-CONTROL-BWA-STRICT-20-GATK-yes	500-CONTROL-BWA-RELAXED-0-GATK-no
500-CONTROL-BWA-STRICT-20-FB-yes	500-CONTROL-BWA-RELAXED-0-FB-no
300-CONTROL-BWA-STRICT-20-GATK-yes	300-CONTROL-BWA-RELAXED-0-GATK-no
150-CONTROL-Bowtie-STRICT-20-GATK-yes	150-CONTROL-Bowtie-RELAXED-0-GATK-no
150-CONTROL-Bowtie-STRICT-20-FB-yes	150-CONTROL-Bowtie-RELAXED-0-FB-no
150-CONTROL-BWA-STRICT-20-GATK-yes	150-CONTROL-BWA-RELAXED-0-GATK-no
100-CONTROL-Bowtie-STRICT-20-GATK-yes	100-CONTROL-Bowtie-RELAXED-0-GATK-no
100-CONTROL-Bowtie-STRICT-20-FB-yes	100-CONTROL-Bowtie-RELAXED-0-FB-no
100-CONTROL-BWA-STRICT-20-GATK-yes	100-CONTROL-BWA-RELAXED-0-GATK-no
50-CONTROL-Bowtie-STRICT-20-GATK-yes	50-CONTROL-Bowtie-RELAXED-0-GATK-no
50-CONTROL-Bowtie-STRICT-20-FB-yes	50-CONTROL-Bowtie-RELAXED-0-FB-no
50-CONTROL-BWA-STRICT-20-GATK-yes	50-CONTROL-BWA-RELAXED-0-GATK-no

Legend: 1,000-50 are read lengths in base pairs; 20-0 are MAPQ settings; STRICT/RELAXED represent mismatch stringency settings; yes/no represent depth-filtering applied or not; FB: FreeBayes

Aiming to provide the most realistic scenario as possible for the experiment, SInC tool (Pattnaik et al., 2014) simulation model was then applied to randomly spike SNP occurrences (in a pre-set rate of 0.0002%) and indels (in a pre-set rate of 0.0001%) on the control reference genome. The transition/transversion ratio used was of 2. SInC tool simulation model is capable of generating both heterozygous and homozygous events in a well (randomly) distributed manner over the genome (Pattnaik et al., 2014). Although the tool is also capable of generating CNVs, these were not generated in this experiment for the sake of simplicity. As a result

of the running of the tool, 23,830 SNPs and 9,134 indels were finally spiked in the ∼120 million base pairs (Mbp) genome.

In order to reproduce the read generation model used in the FP SNPs work, the same SimSeq read simulator (St. John, 2014) was used to generate error-free paired-end reads with 100-fold coverage depth and at lengths of 50, 100, 150, 300, 500, and 1,000 bp from the 'spiked' reference sequence template set originated by SInC tool (following the same planning detailed in Appendix C, subsection C.1.1). Subsequent mapping and variant calling stages were performed utilising the same tools and parameter settings as also detailed in Appendix C and previously in this chapter, having the original control genome as the reference sequence. Table 4.9, below, shows the final SNP numbers obtained (and their breakdown in terms of true positive, false positive, and false negative occurrences) after filtering the called events via the same approach used in the FP SNPs multifactorial experiment (only bi-allelic SNPs considered and with a SNP score higher or equal to 20).

Table 4.9: Final SNP numbers obtained in the FN SNP experiment.

Combinations of the FN SNP experiment and breakdown of the SNP numbers obtained									
Most stringent scenario	#				Most relaxed scenario	#			
Combination	SNPs	TPs	FPs	FNs	Combination	SNPs	TPs	FPs	FNs
1,000-CONTROL-BWA-STRICT-20-GATK-yes	0	0	0	23,830	1,000-CONTROL-BWA-RELAXED-0-GATK-no	25,334	23,595	1,739	235
500-CONTROL-BWA-STRICT-20-GATK-yes	0	0	0	23,830	500-CONTROL-BWA-RELAXED-0-GATK-no	25,244	23,511	1,733	319
500-CONTROL-BWA-STRICT-20-FB-yes	0	0	0	23,830	500-CONTROL-BWA-RELAXED-0-FB-no	24,057	23,752	305	78
300-CONTROL-BWA-STRICT-20-GATK-yes	0	0	0	23,830	300-CONTROL-BWA-RELAXED-0-GATK-no	25,159	23,430	1,729	400
150-CONTROL-Bowtie-STRICT-20-GATK-yes	25,106	23,267	1,839	563	150-CONTROL-Bowtie-RELAXED-0-GATK-no	24,432	22,757	1,675	1,703
150-CONTROL-Bowtie-STRICT-20-FB-yes	24,078	23,257	821	573	150-CONTROL-Bowtie-RELAXED-0-FB-no	24,329	23,745	584	85
150-CONTROL-BWA-STRICT-20-GATK-yes	0	0	0	23,830	150-CONTROL-BWA-RELAXED-0-GATK-no	24,958	23,242	1,716	588
100-CONTROL-Bowtie-STRICT-20-GATK-yes	25,027	23,190	1,837	640	100-CONTROL-Bowtie-RELAXED-0-GATK-no	24,269	22,600	1,669	1,230
100-CONTROL-Bowtie-STRICT-20-FB-yes	25,478	23,187	2,291	643	100-CONTROL-Bowtie-RELAXED-0-FB-no	24,439	23,719	720	111
100-CONTROL-BWA-STRICT-20-GATK-yes	0	0	0	23,830	100-CONTROL-BWA-RELAXED-0-GATK-no	24,818	23,116	1,702	714
50-CONTROL-Bowtie-STRICT-20-GATK-yes	24,838	23,020	1,818	810	50-CONTROL-Bowtie-RELAXED-0-GATK-no	24,041	22,367	1,674	1,463
50-CONTROL-Bowtie-STRICT-20-FB-yes	28,075	23,090	4,985	740	50-CONTROL-Bowtie-RELAXED-0-FB-no	24,720	23,632	1,088	198
50-CONTROL-BWA-STRICT-20-GATK-yes	0	0	0	23,830	50-CONTROL-BWA-RELAXED-0-GATK-no	24,527	22,840	1,687	990

Abbreviations: #: Number of; TPs: true positive SNPs; FPs: false positive SNPs; FNs: false negative SNPs; FB: FreeBayes

Legend: 1,000-50 are read lengths in base pairs; 20-0 are MAPQ settings; STRICT/RELAXED represent mismatch stringency settings; yes/no represent depth-filtering applied or not

From these results, the first immediate observation is that, for all BWA-related 'most stringent' sets, neither true SNPs were recovered nor FP SNPs were generated. This somehow reproduced the behaviour seen in the original FP SNP experiment (Figures 4.4 and 4.5), in which more stringent settings, particularly associated with BWA mapper, yielded 0 FP SNPs, no matter the other factors involved. Therefore, these results suggest that the specific "STRICT-MAPQ20" combination, when applied in conjunction with BWA, is potentially too stringent.

For the remainder combinations, based on Cornish and Guda (2015), the Positive Predictive Value (PPV) (Equation 4.1) and the Sensitivity (Equation 4.2) can be readily retrieved for assessment purposes:

$$PPV = TP/(TP + FP) \tag{4.1}$$

$$Sensitivity = TP/(TP + FN) \tag{4.2}$$

Tables 4.10 and 4.11 show, respectively, the best performing 'non-zeroed' combinations, in terms of PPV and sensitivity, for the FN SNP experiment, computed based on those equations. In this study, PPV ranged from 82.24% to 98.73% while the sensitivity ranged from 93.04% to 99.67%.

Table 4.10: Best performing combinations in terms of PPV for the FN SNP experiment.

Combination	Percentage (%)
500-CONTROL-BWA-RELAXED-0-FB-no	98.73
150-CONTROL-Bowtie-RELAXED-0-FB-no	97.60
100-CONTROL-Bowtie-RELAXED-0-FB-no	97.05
150-CONTROL-Bowtie-STRICT-20-FB-yes	96.59
50-CONTROL-Bowtie-RELAXED-0-FB-no	95.60
150-CONTROL-Bowtie-RELAXED-0-GATK-no	93.14
100-CONTROL-BWA-RELAXED-0-GATK-no	93.14
1,000-CONTROL-BWA-RELAXED-0-GATK-no	93.14
500-CONTROL-BWA-RELAXED-0-GATK-no	93.14
300-CONTROL-BWA-RELAXED-0-GATK-no	93.13
150-CONTROL-BWA-RELAXED-0-GATK-no	93.12
100-CONTROL-Bowtie-RELAXED-0-GATK-no	93.12
50-CONTROL-BWA-RELAXED-0-GATK-no	93.12
50-CONTROL-Bowtie-RELAXED-0-GATK-no	93.04
50-CONTROL-Bowtie-STRICT-20-GATK-yes	92.68
150-CONTROL-Bowtie-STRICT-20-GATK-yes	92.68
100-CONTROL-Bowtie-STRICT-20-GATK-yes	92.66
100-CONTROL-Bowtie-STRICT-20-FB-yes	91.01
50-CONTROL-Bowtie-STRICT-20-FB-yes	82.24

Legend: 1,000-50 are read lengths in base pairs; 20-0 are MAPQ settings; STRICT/RELAXED represent mismatch stringency settings; yes/no represent depth-filtering applied or not; FB: FreeBayes

Table 4.11: Best performing combinations in terms of Sensitivity for the FN SNP experiment.

Combination	Percentage (%)
500-CONTROL-BWA-RELAXED-0-FB-no	99.67
150-CONTROL-Bowtie-RELAXED-0-FB-no	99.64
100-CONTROL-Bowtie-RELAXED-0-FB-no	99.53
50-CONTROL-Bowtie-RELAXED-0-FB-no	99.17
1,000-CONTROL-BWA-RELAXED-0-GATK-no	99.01
500-CONTROL-BWA-RELAXED-0-GATK-no	98.66
300-CONTROL-BWA-RELAXED-0-GATK-no	98.32
150-CONTROL-Bowtie-STRICT-20-GATK-yes	97.64
150-CONTROL-Bowtie-STRICT-20-FB-yes	97.60
150-CONTROL-BWA-RELAXED-0-GATK-no	97.53
100-CONTROL-Bowtie-STRICT-20-GATK-yes	97.31
100-CONTROL-Bowtie-STRICT-20-FB-yes	97.30
100-CONTROL-BWA-RELAXED-0-GATK-no	97.00
50-CONTROL-Bowtie-STRICT-20-FB-yes	96.89
50-CONTROL-Bowtie-STRICT-20-GATK-yes	96.60
50-CONTROL-BWA-RELAXED-0-GATK-no	95.85
100-CONTROL-Bowtie-RELAXED-0-GATK-no	94.84
50-CONTROL-Bowtie-RELAXED-0-GATK-no	93.86
150-CONTROL-Bowtie-RELAXED-0-GATK-no	93.04

Legend: 1,000-50 are read lengths in base pairs; 20-0 are MAPQ settings; STRICT/RELAXED represent mismatch stringency settings; yes/no represent depth-filtering applied or not; FB: FreeBayes

Overall, considering both metrics together, the following general observations can be retrieved from this variant calling experiment using the intact reference sequence for the mappings of reads containing 'known SNPs':

- the more relaxed combinations ('RELAXED-MAPQ0'-related) performed better than their counterparts ('STRICT-MAPQ20'-related);
- three combinations consistently performed better:

  '500-CONTROL-BWA-RELAXED-0-FB-no',

'150-CONTROL-Bowtie-RELAXED-0-FB-no', and

'100-CONTROL-Bowtie-RELAXED-0-FB-no';

- the 500 bp read length apparently presented the best response. Apart from the three cases listed above, in general, there was a slight positive trend in the relationship between read length and a better performance. The 50 bp read length tended to be more prevalent in the worst performing cases;

- Both mappers were represented on the top performing cases although Bowtie seemed to slightly outperform BWA, particularly in terms of PPV metric;

- FreeBayes was more prevalent than GATK in the top performing cases.

In order to expand the investigation regarding the behaviour experienced when BWA mapper was associated to the more stringent settings, BamQC tool (Andrews, 2014) was run over the BAM files generated in this experiment, aiming to inspect the distribution of MAPQ values for all the involved reads belonging to each alignment. In summary, for 'BWA-STRICT' alignments, the highest MAPQ values found were of 5, while for the 'BWA-RELAXED' and Bowtie-related mappings, MAPQ values higher than 20, for instance, can be found (Figures 4.11 and 4.12). Thus, this can explain why SNP calls can be observed in the latter but not in the former cases: MAPQ values under 20 will fail the intrinsic filters of the variant calling tools when associated with the most stringent combinations of this work.

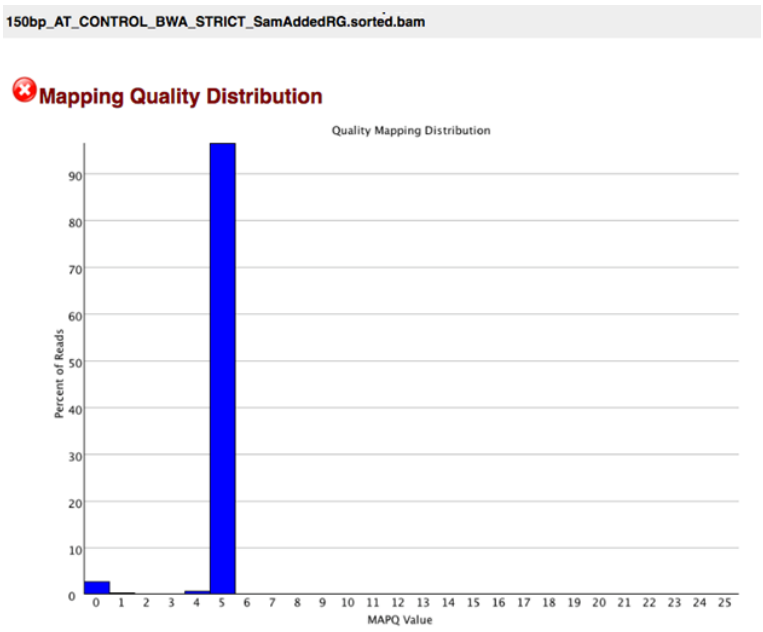

Figure 4.11: BamQC tool screenshot of a 'BWA-STRICT' mapping of the FN SNP experiment.

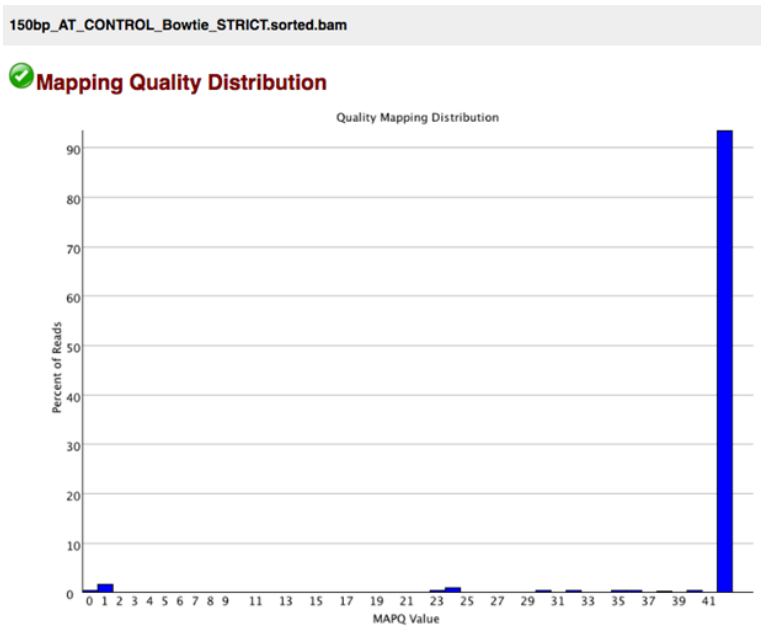

Figure 4.12: BamQC tool screenshot of a 'Bowtie-STRICT' mapping of the FN SNP experiment.

Here, therefore, the large number of FNs in the strict BWA mappings is effectively an artefact and SNPs would be detectable if the MAPQ values assigned by the aligner were not so underestimated. Taking into consideration that the most accurate reference sequence was used in this experiment, in principle, there is no good reason why the MAPQ values should be so low. This would require additional investigation but might be an effect of what was already mentioned in the introductory part of this chapter regarding the mapping quality factor: genuine mappings being under-evaluated by the alignment tools (Ruffalo et al., 2012). It would be interesting, for instance, to have tools like the proposed LoQuM (Ruffalo et al., 2012) added as a factor, in a future multifactorial experiment, in order to verify whether the 'ressurection' of low-labelled mappings would significantly impact the precision of called SNPs.

4.5 Conclusions

The study described here has highlighted and ranked multiple factors that have significant effects on the generation of FP SNPs during variant calling. The quality of the reference sequence was the most important factor. Fragmentation, misassembly, and non-assembly of regions within the reference sequence lead to read mapping targets being effectively unavailable. Corresponding reads then map to incorrect locations, consequently leading to FP SNP accumulation.

Mapping stringency was the second most important factor contributing to FP

SNP numbers in the study, with relaxed mappings generally producing greater numbers of FP SNPs than strict mappings. However, these differences were found to be large only for the combination of Bowtie2, longer reads (300, 500, 1,000 bp) and high quality reference sequence, and BWA with the poor quality reference sequences. This is an important finding, as both the mappers used here are supplied with relatively relaxed mismatch settings as defaults. Thus, running read mappers on relaxed mismatch default settings to maximise the numbers of mapped reads may not be the best approach, especially in cases of a relatively unfinished reference sequence. However, as confirmed in the FN SNP experiment of this work, there is a *caveat* in that very strict mappings may lead to false negative SNPs, and more work is required to formulate an optimal approach to determine a mismatch rate that minimises both false positive and false negative SNPs. Additionally, the work here has proved that, even before attempting to find the best mapping quality filtering criteria, more effort should be put in order to determine a better way of dealing with the potential inconsistency of assigned MAPQ scores.

A complex relationship emerged between read length and FP SNP generation, with the factor playing a comparatively minor role overall. The potential for greater mapping specificity in longer reads is at least partially offset by the increased numbers of mismatches they can contribute, which may be translated into greater

numbers of FP SNPs. Conversely, the threat can be mitigated if a given longer read indeed maps with greater mapping specificity.

The choice of mapper and variant caller also has significant effects upon FP SNP discovery, as does the use of MAPQ and depth filters for SNPs. Between-factor interactions make simple recommendations difficult for a SNP discovery pipeline but, overall, a good quality reference sequence is extremely important for mapping-based variant calling. Stringent mappings and appropriate filtering of SNPs, by at least MAPQ and coverage depth, follow. Here, again, particular care should be taken in order to avoid missing true SNP events. Due to the myriad of simulation tools available nowadays, a potential good practice to adopt before any SNP calling project could be to test different tools, parameters, and filtering settings on subsets of reads/reference sequences emulating the characteristics of the organism/sequencing nature under evaluation. This approach could be beneficial in order to either select or fine tune the best performing combination to be applied in a given project.

Overall, this study's results emphasize the importance of interactions among the factors in a SNP discovery pipeline. The choice of tools and parameters can have a dramatic effect, with particularly poor combinations of software and/or parameter settings yielding tens of thousands of false positives as well as false negatives. Therefore, it is not sufficient just to specify individual parameter values

in isolation, as these can be advantageous or disadvantageous depending upon the choice of other parameter values.

Chapter 5

General conclusions and future work

Preface

This Chapter provides general conclusions of the study and highlights potential future work motivated by it.

5.1 General conclusions

Throughout this PhD project, different experimental designs were used and bioinformatic pipelines were developed to investigate and quantify the mechanisms behind FP SNPs generation.

Paralogs were confirmed as a source of interference with *de novo* assembly process in a potential substantial proportion of cases. Transposable element-derived sequences as well as repeats were also observed as being capable of producing the same issue. This provides strong evidence that the *de novo* assembly process,

irrespective of the assembler used, has created hybrid reference sequences made up from different types of reads that should not be grouped together. This was particularly noticeable with the assembler used to process the simulated DNA-Seq data, where individual bases in the reference sequence were being swapped as a result. This leads to FP SNPs when reads are mapped onto the misassembled reference sequence. Such types of findings could be useful for improving *de novo* assembly tools (e.g. a better tracking process of the reads taking part in a given assembly could flag up occurrences of different classes of reads being erroneously assigned as candidates to solve a particular region of the sequence).

Read mismapping (cross-mapping) due to missing or misassembled reference sequences was also confirmed as a cause of significant numbers of FP SNPs. Repetitive sequences were observed as particularly prone to this kind of misassembly due to the challenges that very similar sequences pose both for *de novo* assembly and read mapping. Multiple factors relating to this were tested and shown to have a significant effect on the generation of FP SNPs in mapping-based SNP discovery. The quality of the reference sequence was the most important, followed by mapping stringency, choice of mapper and variant caller, and filtering of putative SNPs. As a powerful illustration example of the importance of such factors consideration, the work showed that, for the worst performing combination in terms of FP SNPs produced, the numbers obtained could be reduced by 3 orders of magnitude, by

simply changing the mapping stringency factor to the strict setting. The work has also proved, however, that too strict settings may lead to FN SNPs artefacts and that this can happen, for instance, as a side-effect of underestimated mapping quality scores by alignment tools. As another contribution of this PhD project, the read length factor was shown to have a complex relationship with the FP SNPs generation, with two opposing forces interacting: the increased potential for a longer read to cause damage (in terms of the number of mismatches it can contribute) *versus* its increased potential for mapping specificity. In summary, the importance of interactions among the factors involved in a mapping-based SNP discovery pipeline was highlighted, as well as the need for careful selection of combinations of software and/or parameter settings.

5.2 Future work

Potential future works motivated by this thesis project may include:

- Extending the misassembly analysis from Chapter 2 to other assemblers and other organisms. Apart from Velvet, which was used in the investigation of that chapter, other de Bruijn graph-based assemblers are capable of generating the .ACE, .AFG, or .AFG-like files. Examples are MIRA (Chevreux et al., 1999), the Roche 454 Newbler assembler (Margulies et al., 2005), and Ray (Boisvert et al., 2010). Such tools could be options for having their assembly files tracked in a similar way to what was done with the Velvet assembler

in Chapter 2. Irrespective of the assembly file tracking ability, other tools' methods could also be explored (potentially with the alternative strategy of relaxing the mappings also used in the chapter). Tools like Cortex (Iqbal et al., 2012), for instance, utilise variations of the de Bruijn graph, in this case the 'coloured' one. Assembly computation based on the string graph approach is an alternative to the one made with de Bruijn graph method (Henson et al., 2012). Thus, a reference assembled with SGA (Simpson and Durbin, 2012) could be an additional option in such kind of investigation. Independently of the assembly method, different organisms present different genomic characteristics, like poliploidy (Clevenger et al., 2015) and content repetitiveness (Haubold and Wiehe, 2006). Thus, such inherent differences amongst organisms' genomes could be also explored in terms of the impact of the reference sequence misassembly issue by, for instance, expanding the analysis to organisms from other kingdoms along with the specimen from the Plantae one;

- To gain a better understanding of the single-base swap 'glitch' observed in the *de novo* assembler during the DNA-Seq experiment (Chapter 2). This would require, firstly, a deeper analysis of the assembler logs in order to try to identify any particular pattern which could explain such erratic behaviour. An additional approach could involve the usage of just the

reads taking part in the assembly of a problematic contig, aiming to check whether the problem is systematically reproduced or not in a more controlled environment. Another option could be the usage of reads sampled from a different organism, also aiming for the reproducibility of the problem and the posterior analysis of its cause(s);

- To gain a better understanding of the counterintuitive behaviour presented by GATK (increased FP SNP numbers in the callsets) when dealing with MAPQ_20 filter (Chapter 4). Based on what was observed in the FN SNP experiment regarding the inconsistency of mapping quality scores, maybe a broader similar test could be carried out with all the GATK-related combinations (and not only the few ones associated with the control genome) followed by the incorporation of a tool like LoQuM (Ruffalo et al., 2012) in the pipeline. This would potentially reveal whether the trend would be sustained or reversed;

- To gain a better understanding regarding the inferior performance presented by Allpaths assembler, in terms of reference sequence accuracy, when compared to Velvet's one (Chapter 4). A deeper investigation of the causes for such difference in performance between the two assemblers could be carried out, firstly with the specific *A. thaliana* dataset (aiming for reproducibility) (e.g. using varied insert sizes and/or read lengths) and,

secondly, by assembling other organisms' genomes, so the trend could be confirmed or not in a different scenario;

- To expand the analysis of read mismapping (Chapter 4) to include: SGA *de novo* assemblers along with the de Bruijn graph ones; application of parametric and evaluation auxiliary tools/scores like Teaser (Smolka et al., 2015), Genome Mappability Score (GMS) (Lee and Schatz, 2011, 2012), BAYSIC (Cantarel et al., 2014), LoQuM (Ruffalo et al., 2012), etc.; longer read lengths like those provided by Pacific Biosciences and Oxford Nanopore; additional mappers, assemblers, and variant callers as well as additional organisms representing different kingdoms other than the Plantae one. All of these would be categorised as different factors massively expanding the analysis detailed in that chapter;

- To expand the work to false negative SNPs. Firstly, a similar test to the incipient one carried out in Chapter 4 could be redesigned to expand the analysis to all the 576 factor level combinations tested in the original FP SNPs experiment. Posteriorly, the same FN SNP test could be applied in a similar broader analysis like the one proposed in the previous item;

- To gain a better understanding of the spatial distribution of FP SNPs by using quantitative approaches similar to those used in this project (as originally observed by A. Golicz in her unpublished undergraduate honours

project). For instance, a better characterisation of the origin of SNPs around coverage dips and starts/ends of reference sequences (other than reference misassembly and cross-mapping explored here) could be of great interest. The quantifying pipelines developed here could be refactored to automatically explore, categorise, and quantify different spatial patterns of SNP occurrences;

- Establishing an Assemblathon-like competition (Earl et al., 2011) with a view to increasing SNP calling accuracy. Different research teams could be invited for a competition aiming to assess the best SNP calling pipelines. "Golden datasets" of reference sequences as well as simulated and/or real reads would have to be defined prior to the competition, so they could be used for assessing the different solutions proposed by the distinct participating teams. A combined effort like this would surely push the variant calling field forward.

Bibliography

ABI. 2010. The SOLiD^{TM} System: Next-Generation Sequencing. http://www.appliedbiosystems.com/absite/us/en/home/applications-technologies/solid-next-generation-sequencing.html. Accessed: 2016-06-03.

ABMMS. 2014. Genome in a Bottle Consortium. https://sites.stanford.edu/abms/giab. Accessed: 2016-04-24.

Adams, M.D., S.E. Celniker, R.A. Holt, C.A. Evans, J.D. Gocayne, P.G. Amanatides, et al. 2000. The genome sequence of *Drosophila melanogaster*. *Science* 287(5461):2185–2195.

Alkan, C., J.M. Kidd, T. Marques-Bonet, G. Aksay, F. Antonacci, F. Hormozdiari, et al. 2009. Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat Genet* 41(10):1061–1067.

Alkan, C., S. Sajjadian, and E.E. Eichler. 2011. Limitations of next-generation genome sequence assembly. *Nat Meth* 8(1):61–65.

Altmann, A., P. Weber, D. Bader, M. Preuss, E.B. Binder, and B. Müller-Myhsok. 2012. A beginners guide to SNP calling from high-throughput DNA-sequencing data. *Human Genetics* 131(10):1541–1554.

Altschul, S.F., W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. 1990. Basic local alignment search tool. *J Mol Biol* 215(3):403–410.

Anders, S., D.J. McCarthy, Y. Chen, M. Okoniewski, G.K. Smyth, W. Huber, and M.D. Robinson. 2013. Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. *Nat. Protocols* 8(9):1765–1786.

Andrews, S. 2010. FastQC: a quality control tool for high throughput sequence data. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/. Accessed: 2013-03-03.

Andrews, S. 2014. BamQC. https://github.com/s-andrews/BamQC. Acessed: 2016-09-06.

Ansorge, W.J. 1991. Process for sequencing nucleic acids without gel sleving media on solid support and DNA chips (Verfahren zur Sequenzierung von Nukleinsauren ohne Gele). German Patent Application DE 41 41 178 A1 and Corresponding Worldwide Patent Applications.

Ansorge, W.J. 2009. Next-generation DNA sequencing techniques. *New Biotechnology* 25(4):195–203.

Auffray, C., Z. Chen, and L. Hood. 2009. Systems medicine: the future of medical genomics and healthcare. *Genome Medicine* 1(1):2–2.

Babraham Bioinformatics. 2013a. SHERMAN – Bisulfite-Read Simulator v0.1.6. http://www.bioinformatics.babraham.ac.uk/projects/sherman/Sherman_Manual.pdf. Accessed: 2013-10-15.

Babraham Bioinformatics. 2013b. Sherman - bisulfite-treated Read FastQ Simulator. http://www.bioinformatics.babraham.ac.uk/projects/sherman/. Accessed: 2013-10-15.

Bains, W., and G.C. Smith. 1988. A novel method for DNA sequence determination. *Journal of Theoretical Biology* 135:303–307.

Baker, K., M. Bayer, N. Cook, S. Dreißig, T. Dhillon, J. Russell, et al. 2014. The low recombining pericentromeric region of barley restricts gene diversity and evolution but not gene expression. *Plant J* 79(6):981–92.

Baker, M. 2012. *De novo* genome assembly: what every biologist should know. *Nat Meth* 9(4):333–337.

Bao, H., Y. Xiong, H. Guo, R. Zhou, X. Lu, Z. Yang, et al. 2009. MapNext: a software tool for spliced and unspliced alignments and SNP detection of short sequence reads. *BMC Genomics* 10(3):1–6.

Bao, S., R. Jiang, W. Kwan, B. Wang, X. Ma, and Y.Q. Song. 2011. Evaluation of next-generation sequencing software in mapping and assembly. *J Hum Genet* 56(6):406–414.

Barski, A., S. Cuddapah, K. Cui, T.Y. Roh, D.E. Schones, Z. Wang, G. Wei, I. Chepelev, and K. Zhao. 2007. High-resolution profiling of histone methylations in the human genome. *Cell* 129(4):823–837.

bcbio-nextgen. 2015. bcbio. https://bcbio-nextgen.readthedocs.io/en/latest/. Accessed: 2016-04-25.

Beechem, J.M., U. Ulmanello, Y. Wang, M. Yue, M. Lafferty, and H.Y. Sun. 2015. Single Molecule Real-time DNA Sequencing using FRET-based reagents: sequencing DNA on multiple size scales (from single bases to whole chromosomes) to resolve structural variation and enable *de novo* sequencing. https://tools.thermofisher.com/content/sfs/posters/cms_091831.pdf. Accessed: 2016-06-03.

Bennett, S.T. 2004. Solexa Ltd. *Pharmacoeconomics* 5:433–438.

Bennett, S.T., C. Barnes, A. Cox, L. Davies, and C. Brown. 2005. Toward the $1000 human genome. *Pharmacogenomics* 6(4):373–382.

Bentley, D.R. 2006. Whole-genome re-sequencing. *Current Opinion in Genetics & Development* 16(6):545–552. Genomes and evolution.

Bentley, D.R., S. Balasubramanian, H.P. Swerdlow, G.P. Smith, J. Milton, C.G. Brown, et al. 2008. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* 456(7218):53–59.

bioplanet.com. 2013. Genome Comparison and Analytic Testing (GCAT). http://www.bioplanet.com/gcat/. Accessed: 2016-04-24.

Birney, E. 2011. Assemblies: the good, the bad, the ugly. *Nat Meth* 8(1):59–60.

Blue Collar Bioinformatics. 2013. Framework for evaluating variant detection methods: comparison of aligners and callers. https://bcbio.wordpress.com/2013/05/06/framework-for-evaluating-variant-detection-methods-comparison-of-aligners-and-callers/. Accessed: 2014-03-23.

Böcker, S. 2004. Sequencing from compomers: Using mass spectrometry for DNA *de novo* sequencing of 200+ nt. *Journal of Computational Biology* 11(6):1110–1134.

Boisvert, S., F. Laviolette, and J. Corbeil. 2010. Ray: Simultaneous assembly of reads from a mix of high-throughput sequencing technologies. *Journal of Computational Biology* 17(11):1519–1533.

Bolger, A.M., M. Lohse, and B. Usadel. 2014. Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics*. doi:10.1093/bioinformatics/btu170.

Bradnam, K., J.N. Fass, A. Alexandrov, P. Baranay, M. Bechner, I. Birol, et al. 2013. Assemblathon 2: evaluating *de novo* methods of genome assembly in three vertebrate species. *GigaScience* 2(1):10.

Braslavsky, I., B. Hebert, E. Kartalov, and S.R. Quake. 2003. Sequence information can be obtained from single DNA molecules. *Proceedings of the National Academy of Sciences of the United States of America* 100(7):3960–3964.

Bravo, H.C., and R.A. Irizarry. 2010. Model-based quality assessment and base-calling for second-generation sequencing data. *Biometrics* 66(3):665–674.

Breu, Heinz. 2010. A theoretical understanding of 2 base color codes and its application to annotation, error detection, and error correction. `https://www3.appliedbiosystems.com/cms/groups/mcb_marketing/documents/generaldocuments/cms_058265.pdf`. Accessed: 2016-06-03.

Broad Institute. 2012a. GATK. `https://www.broadinstitute.org/gatk/`. Accessed: 2014-11-27.

Broad Institute. 2012b. Performing sequence coverage analysis. `https://www.broadinstitute.org/gatk/guide/article?id=40`. Accessed: 2016-04-24.

Broad Institute. 2014a. Picard. `http://broadinstitute.github.io/picard/`. Accessed: 2014-11-27.

Broad Institute. 2014b. PICARD JDK API documentation. `https://broadinstitute.github.io/picard/javadoc/picard/index.html`. Accessed: 2014-11-27.

Broad Institute. 2014c. Using depth of coverage metrics for variant evaluation. `https://www.broadinstitute.org/gatk/guide/article?id=4721`. Accessed: 2016-04-24.

Broad Institute. 2015a. GATK Best Practices. `https://www.broadinstitute.org/gatk/guide/best-practices.php`. Accessed: 2015-12-20.

Broad Institute. 2015b. Mapping, processing, and duplicate marking with picard tools. `https://www.broadinstitute.org/gatk/events/slides/1503/GATKwh6-BP-1A-Mapping_and_Dedupping.pdf`. Accessed: 2016-04-25.

Brookes, A.J. 1999. The essence of SNPs. *Gene* 234(2):177–186.

Bryant, D.W., Weng-Keen Wong, and T.C. Mockler. 2009. QSRA – a quality-value guided *de novo* short read assembler. *BMC Bioinformatics* 10(1):1–6.

Buffalo, V. 2011. Scythe – A Bayesian adapter trimmer. `http://github.com/vsbuffalo/scythe`. Accessed: 2016-04-20.

Burrows, M., and D.J. Wheeler. 1994. A Block-sorting Lossless Data Compression Algorithm. Tech. Rep., Digital Equipment Corporation – Systems Research Center.

Butler, J., I. MacCallum, M. Kleber, I.A. Shlyakhter, M.K. Belmonte, E.S. Lander, C. Nusbaum, and D.B. Jaffe. 2008. ALLPATHS: *De novo* assembly of whole-genome shotgun microreads. *Genome Research* 18(5):810–820.

Cantarel, B.L., D. Weaver, N. McNeill, J. Zhang, A.J. Mackey, and J. Reese. 2014. BAYSIC: a bayesian method for combining sets of genome variants with improved specificity and sensitivity. *BMC Bioinformatics* 15(1):1–12.

Cao, H., H. Wu, R. Luo, S. Huang, Y. Sun, X. Tong, et al. 2015. *De novo* assembly of a haplotype-resolved human genome. *Nat Biotech* 33(6):617–622.

Chaisson, M.J., D. Brinza, and P.A. Pevzner. 2009. *De novo* fragment assembly with short mate-paired reads: Does the read length matter? *Genome research* 19(2):336–346.

Chaisson, M.J., P. Pevzner, and H. Tang. 2004. Fragment assembly with short reads. *Bioinformatics* 20(13):2067–2074.

Chaisson, M.J., and P.A. Pevzner. 2008. Short read fragment assembly of bacterial genomes. *Genome Research* 18(2):324–330.

Challis, D., J. Yu, U.S. Evani, A.R. Jackson, S. Paithankar, C. Coarfa, A. Milosavljevic, R.A. Gibbs, and F. Yu. 2012. An integrative variant analysis suite for whole exome next-generation sequencing data. *BMC Bioinformatics* 13(1):1–12.

Chang, Z., Z. Wang, and G. Li. 2014. The impacts of read length and transcriptome complexity for *De Novo* assembly: A simulation study. *PLoS ONE* 9(4):e94825.

Chen, Y., T. Souaiaia, and T. Chen. 2009. PerM: efficient mapping of short sequencing reads with periodic full sensitive spaced seeds. *Bioinformatics* 25(19): 2514–2521.

Chevreux, B., T. Wetter, and S. Suhai. 1999. Genome sequence assembly using trace signals and additional sequence information. *Computer Science and Biology: Proceedings of the German Conference on Bioinformatics (GCB)* 45–56.

Cingolani, P., A. Platts, Le Lily Wang, M. Coon, T. Nguyen, L. Wang, S.J. Land, X. Lu, and D.M. Ruden. 2012. A program for annotating and predicting the effects of single nucleotide polymorphisms, snpeff. *Fly* 6(2):80–92. PMID: 22728672.

Clark, S.C., R. Egan, P.I. Frazier, and Z. Wang. 2013. ALE: a generic assembly likelihood evaluation framework for assessing the accuracy of genome and metagenome assemblies. *Bioinformatics* 29(4):435–443.

Clarke, J., Hai-Chen Wu, L. Jayasinghe, A. Patel, S. Reid, and H. Bayley. 2009. Continuous base identification for single-molecule nanopore DNA sequencing. *Nat Nano* 4(4):265–270.

Clevenger, J., C. Chavarro, S.A. Pearl, P. Ozias-Akins, S.A. Jackson, et al. 2015. Single Nucleotide Polymorphism identification in polyploids: A review, example, and recommendations. *Molecular Plant* 8(6):831–846.

CNAG. 2011. dnGASP. `http://cnag.bsc.es/`. Accessed: 2015-05-29.

Cock, P.J.A., C.J. Fields, N. Goto, M.L. Heuer, and P.M. Rice. 2009. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research* 38(6):1767–1771.

Coombs, A. 2008. The sequencing shakeup. *Nat Biotech* 26(10):1109–1112.

Cornish, A., and C. Guda. 2015. A comparison of variant calling pipelines using genome in a bottle as a reference. *BioMed Research International* 2015(456479): 11.

Cornish-Bowden, A. 1985. Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. *Nucleic Acids Research* 13(9): 3021–3030.

Cox, A.J. 2007. ELAND: Efficient large-scale alignment of nucleotide databases. Illumina, San Diego, CA.

Cox, M.P., D.A. Peterson, and P.J. Biggs. 2010. SolexaQA: At-a-glance quality assessment of illumina second-generation sequencing data. *BMC Bioinformatics* 11(1):1–6.

CureFFI.org. 2012. How PCR duplicates arise in next-generation sequencing. http://www.cureffi.org/2012/12/11/how-pcr-duplicates-arise-in-next-generation-sequencing/. Accessed: 2016-04-25.

Dander, A., S. Pabinger, M. Sperk, M. Fischer, G. Stocker, and Z. Trajanoski. 2014. SeqBench: Integrated solution for the management and analysis of exome sequencing data. *BMC Research Notes* 7(1):1–5.

Danecek, P., A. Auton, G. Abecasis, C.A. Albers, E. Banks, M.A. DePristo, B. Handsaker, G. Lunter, G. Marth, S. Sherry, G. McVean, R. Durbin, and 1000 Genomes Project Analysis Group. 2011a. The Variant Call Format and VCFtools. http://vcftools.sourceforge.net/VCF-poster.pdf. Accessed: 2016-06-03.

Danecek, P., A. Auton, G. Abecasis, C.A. Albers, E. Banks, M.A. DePristo, et al. 2011b. The variant call format and VCFtools. *Bioinformatics* 27(15):2156–2158.

Dassanayake, M., Dong-Ha Oh, J.S. Hass, A. Hernandez, H. Hong, S. Ali, et al. 2011. The genome of the extremophile crucifer *Thellungiella parvula*. *Nat Genet* 43(9):913–918.

Davey, J.W., P.A. Hohenlohe, P.D. Etter, J.Q. Boone, J.M. Catchen, and M.L. Blaxter. 2011. Genome-wide genetic marker discovery and genotyping using next-generation sequencing. *Nat Rev Genet* 12(7):499–510.

DePristo, M.A., E. Banks, R. Poplin, K.V. Garimella, J.R. Maguire, C. Hartl, et al. 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat Genet* 43(5):491–498.

Derrien, T., J. Estellé, S. Marco Sola, D.G. Knowles, E. Raineri, R. Guigó, and P. Ribeca. 2012. Fast computation and applications of genome mappability. *PLoS ONE* 7(1):1–16.

Deschamps, S., and M.A. Campbell. 2009. Utilization of next-generation sequencing platforms in plant genomics and genetic variant discovery. *Molecular Breeding* 25(4):553–570.

Dhanapal, A. 2012. Genomics of crop plant genetic resources. *Advances in Bioscience and Biotechnology* 3:378–385.

Dohm, J.C., C. Lottaz, T. Borodina, and H. Himmelbauer. 2007. SHARCGS, a fast and highly accurate short-read assembly algorithm for *de novo* genomic sequencing. *Genome Research* 17(11):1697–1706.

Dohm, J.C., C. Lottaz, T. Borodina, and H. Himmelbauer. 2008. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Research* 36(16):e105.

Dou, J., X. Zhao, X. Fu, W. Jiao, N. Wang, L. Zhang, X. Hu, S. Wang, and Z. Bao. 2012. Reference-free SNP calling: improved accuracy by preventing incorrect calls from repetitive genomic regions. *Biology Direct* 7(1):17.

Dressman, D., H. Yan, G. Traverso, K.W. Kinzler, and B. Vogelstein. 2003. Transforming single DNA molecules into fluorescent magnetic particles for detection and enumeration of genetic variations. *Proceedings of the National Academy of Sciences* 100(15):8817–8822.

Drmanac, R., I. Labat, I. Brukner, and R. Crkvenjakov. 1989. Sequencing of megabase plus DNA by hybridization: theory of the method. *Genomics* 4: 114–28.

Droege, M., and B. Hill. 2008. The Genome Sequencer FLX system – Longer reads, more applications, straight forward bioinformatics and more complete data sets. *Journal of Biotechnology* 136(12):3–10. Genome Research in the Light of Ultrafast Sequencing Technologies.

Duran, C., N. Appleby, D. Edwards, and J. Batley. 2009. Molecular genetic markers: Discovery, applications, data storage and visualisation. *Current Bioinformatics* 4(1):16–27.

Earl, D., K. Bradnam, J. St. John, A. Darling, D. Lin, J. Fass, et al. 2011. Assemblathon 1: A competitive assessment of de novo short read assembly methods. *Genome Research*. doi:10.1101/gr.126599.111.

Edwards, D., and J. Batley. 2010. Plant genome sequencing: applications for crop improvement. *Plant Biotechnology Journal* 8(1):2–9.

Edwards, M. A., and R. J. Henry. 2011. DNA sequencing methods contributing to new directions in cereal research. *Journal of Cereal Science* 54(3):395 – 400.

Eid, J., A. Fehr, J. Gray, K. Luong, J. Lyle, G. Otto, et al. 2009. Real-time DNA sequencing from single polymerase molecules. *Science* 323(5910):133–138.

EMBL-EBI. 2008. Manual.pdf (velvet manual - version 1.1). `https://www.ebi.ac.uk/~zerbino/velvet/Manual.pdf`. Accessed: 2013-10-15.

EMBL-EBI. 2012. HTS Mappers. `http://www.ebi.ac.uk/~nf/hts_mappers/`. Accessed: 2016-04-25.

Emboss. 1999. emboss revseq. http://emboss.sourceforge.net/apps/release/6.3/emboss/apps/revseq.html. Accessed: 2014-11-20.

Ensembl. 2016. About Ensembl Variation. http://www.ensembl.org/info/genome/variation/index.html. Accessed: 2016-06-06.

Ewing, B., and P. Green. 1998. Base-calling of automated sequencer traces using Phred. II. Error probabilities. *Genome Research* 8(3):186–194.

Ewing, B., LaDeana Hillier, M.C. Wendl, and P. Green. 1998. Base-calling of automated sequencer traces using Phred.I. Accuracy assessment. *Genome Research* 8(3):175–185.

Farrer, R.A., D.A. Henk, D. MacLean, D.J. Studholme, and M.C. Fisher. 2013. Using False Discovery Rates to benchmark SNP-callers in next-generation sequencing projects. *Sci Rep* 3. doi:10.1038/srep01512.

Fitch, W.M. 1970. Distinguishing homologous from analogous proteins. *Systematic Biology* 19(2):99–113.

Fitch, W.M. 2000. Homology: a personal view on some of the problems. *Trends in Genetics* 16(5):227–231.

Flicek, P., and E. Birney. 2009. Sense from sequence reads: methods for alignment and assembly. *Nat Meth* 6(11s):S6–S12.

Fonseca, N.A., J. Rung, A. Brazma, and J.C. Marioni. 2012. Tools for mapping high-throughput sequencing data. *Bioinformatics* 28(24):3169–3177.

G2P. 2010. Tools predicting the overall functional consequences of SNPs. http://gen2phen.org/wiki/tools-predicting-overal-functional-consequences-snps. Accessed: 2016-04-25.

GAGE. 2011. Genome Assembly Gold-standard Evaluations. http://gage.cbcb.umd.edu/assemblers/index.html. Accessed: 2015-05-29.

Garrison, E. 2012. freebayes. https://github.com/ekg/freebayes. Accessed: 2013-10-15.

Garrison, E. 2013. vcflib. https://github.com/ekg/vcflib. Accessed: 2013-10-15.

Garrison, E. 2014. naïve variant calling. https://groups.google.com/forum/\#!topic/freebayes/_PufWT29eoE. Accessed: 2014-03-23.

Garrison, E., and G. Marth. 2012. Haplotype-based variant detection from short-read sequencing. arxiv:1207.3907.

Gilles, A., E. Meglécz, N. Pech, S. Ferreira, T. Malausa, and Jean-François Martin. 2011. Accuracy and quality assessment of 454 GS-FLX Titanium pyrosequencing. *BMC Genomics* 12(1):1–11.

gkno. 2013. gkno. `http://gkno.me/`. Accessed: 2016-04-24.

Glenn, T.C. 2011. Field guide to next-generation DNA sequencers. *Molecular Ecology Resources* 11(5):759–769.

Gnerre, S., I. MacCallum, D. Przybylski, F.J. Ribeiro, J.N. Burton, B.J. Walkerand, et al. 2011. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences* 108(4):1513–1518.

Goldstein, D.B., and G.L. Cavalleri. 2005. Genomics: Understanding human diversity. *Nature* 437(7063):1241–1242.

Golicz, A., P.A. Martinez, M. Zander, D.A. Patel, A.P. Van De Wouw, P. Visendi, T.L. Fitzgerald, D. Edwards, J. Batley, et al. 2014. Gene loss in the fungal canola pathogen *Leptosphaeria maculans*. *Functional & Integrative Genomics* 15(2):189–196.

Grabherr, M.G., B.J. Haas, M. Yassour, J.Z. Levin, D.A. Thompson, I. Amit, et al. 2011. Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nat Biotech* 29(7):644–652.

Griffiths, A.J.F., S.R. Wessler, S.B. Carroll, and J. Doebley. 2012. *Introduction to genetic analysis*. 10th ed. W. H. Freeman and Company.

Guffanti, A., M. Iacono, P. Pelucchi, N. Kim, G. Soldà, L.J. Croft, et al. 2009. A transcriptional sketch of a primary human breast cancer by 454 deep sequencing. *BMC Genomics* 10(1):1–17.

Gurevich, A., V. Saveliev, N. Vyahhi, and G. Tesler. 2013. QUAST: quality assessment tool for genome assemblies. *Bioinformatics (Oxford, England)* 29(8): 1072–1075.

Haas, B.J., A. Papanicolaou, M. Yassour, M. Grabherr, P.D. Blood, J. Bowden, et al. 2013. *De novo* transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. *Nat. Protocols* 8(8): 1494–1512.

Hamilton, J.P., and C.R. Buell. 2012. Advances in plant genome sequencing. *The Plant Journal* 70(1):177–190.

Hannon, G. 2009. FASTX-Toolkit. `hannonlab.cshl.edu/fastx_toolkit/`. Accessed: 2016-04-20.

HapMap. 2003. International Hapmap Project. `http://www.hapmap.org`. Accessed: 2016-03-06.

Harismendy, O., P.C. Ng, R.L. Strausberg, X. Wang, T.B. Stockwell, K.Y. Beeson, et al. 2009. Evaluation of next generation sequencing platforms for population targeted sequencing studies. *Genome Biology* 10(3):1–13.

Hartl, D.L. 2011. *Essential genetics: A genomics perspective*. Jones and Bartlett Publishers, LLC.

Hatem, A., D. Bozdağ, A.E. Toland, Ü.V. Çatalyürek, et al. 2013. Benchmarking short sequence mapping tools. *BMC Bioinformatics* 14(1):1–25.

Haubold, Bernhard, and Thomas Wiehe. 2006. How repetitive are genomes? *BMC Bioinformatics* 7(1):1–10.

Henry, R., and K. Edwards. 2009. New tools for single nucleotide polymorphism (SNP) discovery and analysis accelerating plant biotechnology. *Plant Biotechnology Journal* 7(4):311–311.

Henson, J., G. Tischler, and Z. Ning. 2012. Next-generation sequencing and large genome assemblies. *Pharmacogenomics* 13(8):901–915.

Hernandez, D., P. François, L. Farinelli, M. Østerås, and J. Schrenzel. 2008. *De novo* bacterial genome sequencing: Millions of very short reads assembled on a desktop computer. *Genome Research* 18(5):802–809.

Homer, N., B. Merriman, S.F. Nelson, et al. 2009. BFAST: An alignment tool for large scale genome resequencing. *PLoS ONE* 4(11):1–12.

Horner, D.S., G. Pavesi, T. Castrignanò, P. D'Onorio De Meo, S. Liuni, M. Sammeth, E. Picardi, G. Pesole, et al. 2010. Bioinformatics approaches for genomics and post genomics applications of next-generation sequencing. *Briefings in Bioinformatics* 11(2):181–197.

Huang, X., and A. Madan. 1999. CAP3: A DNA sequence assembly program. *Genome Research* 9(9):868–877. `http://genome.cshlp.org/content/9/9/868.full.pdf+html`.

Hutchison, C.A. 3rd. 2007. DNA sequencing: bench to bedside and beyond. *Nucleic Acids Research* 35(18):6227–6237.

Hyman, E.D. 1988. A new method of sequencing DNA. *Analytical Biochemistry* 174:423–36.

IBGSC, The International Barley Genome Sequencing Consortium. 2012. A physical, genetic and functional sequence assembly of the barley genome. *Nature* 491(7426):711–716.

Illumina, Inc. 2009. Systems. `http://www.illumina.com/systems.ilmn`. Accessed: 2014-05-29.

Illumina, Inc. 2014a. HiSeq System Performance Parameters. `http://systems.illumina.com/systems/hiseq_2500_1500/performance_specifications.ilmn`. Accessed: 2014-10-18.

Illumina, Inc. 2014b. Intro to Sequencing by Synthesis: Industry-leading Data Quality. `http://www.youtube.com/embed/HMyCqWhwB8E?iframe&rel=0&autoplay=1`. Accessed: 2016-04-23.

Imelfort, M., C. Duran, J. Batley, and D. Edwards. 2009. Discovering genetic polymorphisms in next-generation sequencing data. *Plant Biotechnology Journal* 7(4):312–317.

Iqbal, Z., M. Caccamo, I. Turner, P. Flicek, and G. McVean. 2012. *De novo* assembly and genotyping of variants using colored de Bruijn graphs. *Nat Genet* 44(2):226–232.

IWGSC, The International Wheat Genome Sequencing Consortium. 2014. A chromosome-based draft sequence of the hexaploid bread wheat (*Triticum aestivum*) genome. *Science* 345(6194):1251788.

Jeck, W.R., J.A. Reinhardt, D.A. Baltrus, M.T. Hickenbotham, V. Magrini, E.R. Mardis, J.L. Dangl, and C.D. Jones. 2007. Extending assembly of short DNA sequences to handle error. *Bioinformatics* 23(21):2942–2944.

Jensen, R.A. 2001. Orthologs and paralogs – we need to get it right. *Genome Biology* 2(8):interactions1002.1–interactions1002.3.

JGI. 1997. Statistics. `https://gold.jgi-psf.org/statistics`. Accessed: 2014-10-18.

Jiang, H., and W.H. Wong. 2008. SeqMap: mapping massive amount of oligonucleotides to the genome. *Bioinformatics* 24(20):2395–2396.

Johns Hopkins University. 2009. Bowtie – An ultrafast memory-efficient short read aligner – Table of Contents. http://bowtie-bio.sourceforge.net/manual.shtml. Accessed: 2016-06-03.

Johns Hopkins University. 2014. Bowtie 2 – Fast and sensitive read alignment – Table of Contents. http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml. Accessed: 2014-05-15.

Johnson, D.S., A. Mortazavi, R.M. Myers, and B. Wold. 2007. Genome-wide mapping of in vivo protein-DNA interactions. *Science* 316(5830):1497–1502.

Joshi, N.A., and J.N. Fass. 2011. Sickle: A sliding-window, adaptive, quality-based trimming tool for FastQ files. https://github.com/najoshi/sickle. Accessed: 2016-04-20.

Kahvejian, A., J. Quackenbush, and J.F. Thompson. 2008. What would you do if you could sequence everything? *Nat Biotech* 26(10):1125–1133.

Kao, Wei-Chun, and Y.S. Song. 2011. naiveBayesCall: An efficient model-based base-calling algorithm for High-Throughput Sequencing. *Journal of Computational Biology* 18(3):365–377.

Kao, Wei-Chun, K. Stevens, and Y.S. Song. 2009. BayesCall: A model-based base-calling algorithm for high-throughput short-read sequencing. *Genome Research* 19(10):1884–1895.

Kent, W.J. 2002. BLAT - the BLAST-Like Alignment Tool. *Genome Research* 12(4):656–664.

Kircher, M., U. Stenzel, and J. Kelso. 2009. Improved base calling for the Illumina Genome Analyzer using machine learning strategies. *Genome Biology* 10(8):1–9.

Kling, J. 2005. The search for a sequencing thoroughbred. *Nat Biotech* 23(11):1333–1335.

Koboldt, D.C., K. Chen, T. Wylie, D.E. Larson, M.D. McLellan, E.R. Mardis, G.M. Weinstock, R.K. Wilson, and L. Ding. 2009. VarScan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics* 25(17):2283–2285.

Krueger, F., and S.R. Andrews. 2012. Quality control, trimming and alignment of Bisulfite-Seq data (Prot 57). http://www.epigenesys.eu/en/protocols/bio-informatics/483-quality-control-trimming-and-alignment-of-bisulfite-\\seq-data-prot-57. Epigenesys protocol.

Krueger, F., B. Kreck, A. Franke, and S.R. Andrews. 2012. DNA methylome analysis using short bisulfite sequencing data. *Nat Meth* 9(2):145–151.

Kumar, S., T.W. Banks, and S. Cloutier. 2012. SNP discovery through Next-Generation Sequencing and its applications. *Int J Plant Genomics* 2012(831460):1–15.

Kunda, H. 2015. Benchmarking current NGS mapping algorithms. `http://gbsc.stanford.edu/docs/HemantKunda2015.pdf`.

Kuzniar, A., R.C. van Ham, S. Pongor, and J.A. Leunissen. 2008. The quest for orthologs: finding the corresponding gene across genomes. *Trends in Genetics* 24(11):539–551.

Lai, K., C. Duran, P.J. Berkman, M.T. Lorenc, J. Stiller, S. Manoli, et al. 2012. Single nucleotide polymorphism discovery from wheat next-generation sequence data. *Plant Biotechnol J* 10(6):743–749.

Laird, P.W. 2010. Principles and challenges of genome-wide DNA methylation analysis. *Nat Rev Genet* 11(3):191–203.

Lam, H.Y.K., C. Pan, M.J. Clark, P. Lacroute, R. Chen, R. Haraksingh, et al. 2012. Detecting and annotating genetic variations using the HugeSeq pipeline. *Nat Biotech* 30(3):226–229.

Langmead, B. 2014. De Bruijn Graph assembly. `http://www.cs.jhu.edu/~langmea/resources/lecture_notes/assembly_dbg.pdf`. Accessed: 2016-02-13.

Langmead, B., and S.L. Salzberg. 2012. Fast gapped-read alignment with Bowtie 2. *Nat Meth* 9(4):357–359.

Langmead, B., C. Trapnell, M. Pop, and S. Salzberg. 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology* 10(3):R25.

Lee, H., and M.C. Schatz. 2012. Genomic dark matter: the reliability of short read mapping illustrated by the genome mappability score. *Bioinformatics* 28(16): 2097–2105.

Lee, Hayan, and Michael C. Schatz. 2011. Genomic Dark Matter: The reliability of short read mapping illustrated by the Genome Mappability Score (GMS). `http://schatzlab.cshl.edu/publications/posters/2011.PersonalGenomes.Dark_Matter.pdf`. Accessed: 2016-04-24.

Lee, Wan-Ping, M.P. Stromberg, A. Ward, C. Stewart, E.P. Garrison, and G.T. Marth. 2014. Mosaik: A hash-based algorithm for accurate next-generation sequencing short-read mapping. *PLoS ONE* 9(3):1–11.

Leggett, R.M., and D. MacLean. 2014. Reference-free SNP detection: dealing with the data deluge. *BMC Genomics* 15(4):1–7.

Leggett, R.M., R.H. Ramirez-Gonzalez, W. Verweij, C.G. Kawashima, Z. Iqbal, J.D.G. Jones, M. Caccamo, and D. MacLean. 2013. Identifying and classifying trait linked polymorphisms in non-reference species by walking coloured de Bruijn graphs. *PLoS ONE* 8(3):e60058.

Li, H. 2013. Manual Reference Pages – bwa (1). `http://bio-bwa.sourceforge.net/bwa.shtml`. Accessed: 2014-11-20.

Li, H. 2014a. fermi – a WGS de novo assembler based on the FMD-index for large genomes. `https://github.com/lh3/fermi`. Accessed: 2016-06-14.

Li, H. 2014b. Towards better understanding of artifacts in variant calling from high-coverage samples.arxiv:1404.0929v1.

Li, H., and R. Durbin. 2009. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25(14):1754–1760.

Li, H., and R. Durbin. 2010. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics* 26(5):589–595.

Li, H., B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, et al. 2009a. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25(16): 2078–2079.

Li, H., and N. Homer. 2010. A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in Bioinformatics* 11(5):473–483.

Li, H., J. Ruan, and R. Durbin. 2008a. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research* 18(11): 1851–1858.

Li, H., et al. 2012. SAMtools. `http://samtools.sourceforge.net/`. Accessed: 2013-04-01.

Li, R., Y. Li, X. Fang, H. Yang, Jian Wang, K. Kristiansen, and Jun Wang. 2009b. SNP detection for massively parallel whole-genome resequencing. *Genome Research* 19(6):1124–1132.

Li, R., Y. Li, K. Kristiansen, and J. Wang. 2008b. SOAP: short oligonucleotide alignment program. *Bioinformatics* 24(5):713–714.

Li, R., C. Yu, Y. Li, Tak-Wah Lam, Siu-Ming Yiu, K. Kristiansen, and J. Wang. 2009c. SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics* 25(15):1966–1967.

Li, Ruiqiang, W. Fan, G. Tian, H. Zhu, L. He, J. Cai, et al. 2010. The sequence and *de novo* assembly of the giant panda genome. *Nature* 463(7279):311–317.

Li, Ruiqiang, Hongmei Zhu, Jue Ruan, Wubin Qian, Xiaodong Fang, Zhongbin Shi, Yingrui Li, Shengting Li, Gao Shan, Karsten Kristiansen, Songgang Li, Huanming Yang, Jian Wang, and Jun Wang. 2009d. *De novo* assembly of human genomes with massively parallel short read sequencing. *Genome Research* 20(2): 265–272.

Liao, P.-Y., and K.H. Lee. 2010. From SNPs to functional polymorphism: The insight into biotechnology applications. *Biochem Eng J* 49(2):149–158.

Lin, H., Z. Zhang, M.Q. Zhang, B. Ma, and M. Li. 2008. ZOOM! Zillions of oligos mapped. *Bioinformatics* 24(21):2431–2437.

Liu, Lin, Yinhu Li, Siliang Li, Ni Hu, Yimin He, Ray Pong, Danni Lin, Lihua Lu, and Maggie Law. 2012. Comparison of Next-Generation Sequencing systems. *Journal of Biomedicine and Biotechnology* 2012:11. Article ID 251364.

Liu, X., S. Han, Z. Wang, J. Gelernter, and Bao-Zhu Yang. 2013. Variant callers for Next-Generation Sequencing data: A comparison study. *PLoS ONE* 8(9): e75619.

Lorenc, M., S. Hayashi, J. Stiller, H. Lee, S. Manoli, P. Ruperao, et al. 2012. Discovery of Single Nucleotide Polymorphisms in complex genomes using SGSautoSNP. *Biology* 1(2):370–382.

Lu, Z.H., A. L. Archibald, and T. Ait-Ali. 2014. Beyond the whole genome consensus: Unravelling of PRRSV phylogenomics using next generation sequencing technologies. *Virus Research* 194:167–174. Nidoviruses I.

Luckey, J.A., H. Drossman, A.J. Kostichka, D.A. Mead, J. D'Cunha, T.B. Norris, and L.M. Smith. 1990. High speed DNA sequencing by capillary electrophoresis. *Nucleic Acids Research* 18(15):4417–4421.

Lunter, G., and M. Goodson. 2011. Stampy: A statistical algorithm for sensitive and fast mapping of Illumina sequence reads. *Genome Research* 21(6):936–939.

Maestre, H.L., L. Brinza, C. Marchet, J. Kielbassa, S. Bastien, M. Boutignyand, et al. 2015. *De novo* identification, differential analysis and functional annotation of SNPs from RNA-seq data in non-model species. *bioRxiv*. doi:10.1101/035238.

Manske, H.M., and D.P. Kwiatkowski. 2009. SNP-o-matic. *Bioinformatics* 25(18): 2434–2435.

Mardis, E.R. 2008. The impact of next-generation sequencing technology on genetics. *Trends in Genetics* 24(3):133–141.

Mardis, E.R. 2010. The $1,000 genome, the $100,000 analysis? *Genome Medicine* 2(11):1–3.

Mardis, E.R. 2011. A decade's perspective on DNA sequencing technology. *Nature* 470(7333):198–203.

Mardis, E.R. 2013. Next-Generation Sequencing platforms. *Annual Review of Analytical Chemistry* 6(1):287–303. PMID: 23560931.

Margulies, M., M. Egholm, W.E. Altman, S. Attiya, J.S. Bader, L.A. Bemben, et al. 2005. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 437(7057):376–380.

Martin, E.R., D.D. Kinnamon, M.A. Schmidt, E.H. Powell, S. Zuchner, and R.W. Morris. 2010. SeqEM: an adaptive genotype-calling approach for next-generation sequencing studies. *Bioinformatics* 26(22):2803–2810.

Martin, J.A., and Z. Wang. 2011. Next-generation transcriptome assembly. *Nat Rev Genet* 12(10):671–682.

Martin, M. 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal* 17(1):10–12.

Matsumoto, T., T. Tanaka, H. Sakai, N. Amano, H. Kanamori, K. Kurita, et al. 2011. Comprehensive sequence analysis of 24,783 barley full-length cDNAs derived from 12 clone libraries. *Plant Physiology* 156(1):20–28.

Maxam, A M, and W Gilbert. 1977. A new method for sequencing DNA. *Proceedings of the National Academy of Sciences of the United States of America* 74(2):560–564.

McKenna, A., M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, et al. 2010. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research* 20(9): 1297–1303.

Meacham, F., D. Boffelli, J. Dhahbi, D.I.K. Martin, M. Singer, and L. Pachter. 2011. Identification and correction of systematic error in high-throughput sequence data. *BMC Bioinformatics* 12(1):1–11.

Medvedev, P., M. Stanciu, and M. Brudno. 2009. Computational methods for discovering structural variation with next-generation sequencing. *Nat Meth* 6(11s):S13–S20.

Metzker, M.L. 2005. Emerging technologies in DNA sequencing. *Genome Research* 15(12):1767–1776.

Metzker, M.L. 2010. Sequencing technologies – the next generation. *Nat Rev Genet* 11(1):31–46.

Miller, J.R., A.L. Delcher, S. Koren, E. Venter, B.P. Walenz, A. Brownley, J. Johnson, K. Li, C. Mobarry, and G. Sutton. 2008. Aggressive assembly of pyrosequencing reads with mates. *Bioinformatics* 24(24):2818–2824.

Miller, J.R., S. Koren, and G. Sutton. 2010. Assembly algorithms for next-generation sequencing data. *Genomics* 95(6):315–327.

Milne, I., M. Bayer, L. Cardle, P. Shaw, G. Stephen, F. Wright, and D. Marshall. 2010. Tablet – next generation sequence assembly visualization. *Bioinformatics* 26(3):401–402.

Milne, I., G. Stephen, M. Bayer, P.J.A. Cock, L. Pritchard, L. Cardle, P. Shaw, and D. Marshall. 2013a. Using Tablet for visual exploration of second-generation sequencing data. *Briefings in bioinformatics* 14(2):193–202.

Milne, I., G. Stephen, M. Bayer, C. Linda, P. Shah, and D. Marshall. 2013b. Visual validation of NGS data features using Tablet. Plant and Animal Genome XXI, 2013. San Diego.

Mitra, R.D., J. Shendure, J. Olejnik, Edyta-Krzymanska-Olejnik, and G.M. Church. 2003. Fluorescent *in situ* sequencing on polymerase colonies. *Analytical Biochemistry* 320:55–65.

Morin, P.A., G. Luikart, R.K. Wayne, and the SNP workshop group. 2004. SNPs in ecology, evolution and conservation. *Trends Ecol Evol* 19(4):208–216.

Morozova, Olena, and Marco A. Marra. 2008. Applications of next-generation sequencing technologies in functional genomics. *Genomics* 92(5):255–264.

Myers, E.W. 1995. Toward simplifying and accurately formulating fragment assembly. *Journal of Computational Biology* 2(2):275–290.

Myers, E.W., G.G. Sutton, A.L. Delcher, I.M. Dew, D.P. Fasulo, M.J. Flanigan, et al. 2000. A whole-genome assembly of *Drosophila*. *Science* 287(5461): 2196–2204.

Myles, S., Jer-Ming Chia, B. Hurwitz, C. Simon, G.Y. Zhong, E. Buckler, and D. Ware. 2010. Rapid genomic characterization of the genus *Vitis*. *PLoS ONE* 5(1):1–9.

Nakamura, K., T. Oshima, T. Morimoto, S. Ikeda, H. Yoshikawa, Y. Shiwa, et al. 2011. Sequence-specific error profile of Illumina sequencers. *Nucleic Acids Research* 39(13):e90.

NCBI. 1998. dbSNP – Short Genetic Variation. `http://www.ncbi.nlm.nih.gov/SNP/`. Accessed: 2016-04-25.

Nielsen, R., J.S. Paul, A. Albrechtsen, and Y.S. Song. 2011. Genotype and SNP calling from next-generation sequencing data. *Nat Rev Genet* 12(6):443–451.

NIH. 2001. Arabidopsis. `http://www.nih.gov/science/models/arabidopsis/`. Accessed: 2013-10-15.

NIH. 2007. What are single nucleotide polymorphisms (SNPs)? `https://ghr.nlm.nih.gov/handbook/genomicresearch/snp`. Accessed: 2016-05-31.

Ning, Z., A.J. Cox, and J.C. Mullikin. 2001. SSAHA: A fast search method for large DNA databases. *Genome Research* 11(10):1725–1729.

Novocraft. 2008. Novoalign. `http://www.novocraft.com`. Accessed: 2016-04-25.

Nyrén, P., and A. Lundin. 1985. Enzymatic method for continuous monitoring of inorganic pyrophosphate synthesis. *Analytical Biochemistry* 151:504–9.

Nyrén, P., B. Pettersson, and M. Uhlen. 1993. Solid phase DNA minisequencing by an enzymatic luminometric inorganic pyrophosphate detection assay. *Analytical Biochemistry* 208(1):171–175.

Nystedt, B., N.R. Street, A. Wetterbom, A. Zuccolo, Yao-Cheng Lin, D.G. Scofield, et al. 2013. The Norway spruce genome sequence and conifer genome evolution. *Nature* 497(7451):579–584.

Ondov, B.D., A. Varadarajan, K.D. Passalacqua, and N.H. Bergman. 2008. Efficient mapping of Applied Biosystems SOLiD sequence data to a reference genome for functional genomic applications. *Bioinformatics* 24(23):2776–2777.

O'Rawe, J., T. Jiang, G. Sun, Y. Wu, W. Wang, J. Huand, et al. 2013. Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing. *Genome Medicine* 5(3):28.

Otto, C., P.F. Stadler, and S. Hoffmann. 2014. Lacking alignments? The next-generation sequencing mapper segemehl revisited. *Bioinformatics* 30(13): 1837–1843.

Oxford Nanopore Technologies. 2008. Oxford Nanopore Technologies. `https://www.nanoporetech.com/`. Accessed: 2015-07-31.

Pabinger, S., A. Dander, M. Fischer, R. Snajder, M. Sperk, M. Efremova, B. Krabichler, M.R. Speicher, J. Zschocke, and Z. Trajanoski. 2014. A survey of tools for variant analysis of next-generation genome sequencing data. *Briefings in Bioinformatics* 15(2):256–278.

Pacific Biosciences of California, Inc. 2015. PACBIO. `http://www.pacb.com/`. Accessed: 2015-07-31.

Paszkiewicz, K., and D.J. Studholme. 2010. *De novo* assembly of short sequence reads. *Briefings in Bioinformatics* 11(5):457–472.

Pattnaik, S., S. Gupta, A. A. Rao, and B. Panda. 2014. SInC: an accurate and fast error-model based simulator for SNPs, indels and CNVs coupled with a read generator for short-read sequence data. *BMC Bioinformatics* 15:40–40.

Payne, R., D. Murray, S. Harding, D. Baird, and D. Soutar. 2013. *Introduction to GenStat for Windows*. VSN International, Hemel Hempstead, 16th ed.

Peng, X., J. Wang, Z. Zhang, Q. Xiao, M. Li, and Y. Pan. 2015. Re-alignment of the unmapped reads with base quality score. *BMC Bioinformatics* 16(Suppl 5): S8.

Pettersson, E., J. Lundeberg, and A. Ahmadian. 2009. Generations of sequencing technologies. *Genomics* 93(2):105–111.

Phillippy, A., M. Schatz, and M. Pop. 2008. Genome assembly forensics: finding the elusive mis-assembly. *Genome Biology* 9(3):R55.

Poland, J.A., and T.W. Rife. 2012. Genotyping-by-Sequencing for plant breeding and genetics. *Plant Biotechnol J* 5(3):92–102.

Polonsky, S., G. Stolovitzky, and S. Rossnagel. 2007. DNA transistor. Research Report RC24242 (W0704-094), IBM Research Division, Thomas J. Watson Research Center - P.O. Box 218 - Yorktown Heights, NY 10598. Other.

Pop, M. 2009. Genome assembly reborn: recent computational challenges. *Briefings in Bioinformatics* 10(4):354–366.

Pop, M., and S.L. Salzberg. 2008. Bioinformatics challenges of new sequencing technology. *Trends in genetics : TIG* 24(3):142–149.

Preparata, F.P., and E. Upfal. 2000. Sequencing-by-Hybridization at the information-theory bound: An optimal algorithm. *Journal of Computational Biology* 7(3-4):621–630.

Prober, J.M., G.L. Trainor, R.J. Dam, F.W. Hobbs, C.W. Robertson, R.J. Zagursky, A.J. Cocuzza, M.A. Jensen, and K. Baumeister. 1987. A system for rapid DNA sequencing with fluorescent chain-terminating dideoxynucleotides. *Science* 238(4825):336–341.

Qin, J., R. Li, J. Raes, M. Arumugam, K.S. Burgdorf, C. Manichanhand, et al. 2010. A human gut microbial gene catalogue established by metagenomic sequencing. *Nature* 464(7285):59–65.

Quinlan, A.R., D.A. Stewart, M.P. Stromberg, and G.T. Marth. 2008. Pyrobayes: an improved base caller for SNP discovery in pyrosequences. *Nat Meth* 5(2): 179–181.

Ratan, A., Y. Zhang, V.M. Hayes, S.C. Schuster, and W. Miller. 2010. Calling SNPs without a reference sequence. *BMC Bioinformatics* 11(1):1–13.

Ribeiro, A., A. Golicz, C.A. Hackett, I. Milne, G. Stephen, D. Marshall, A.J. Flavell, and M. Bayer. 2015. An investigation of causes of false positive single nucleotide polymorphisms using simulated reads from a small eukaryote genome. *BMC Bioinformatics* 16(1):1–16.

Ribeiro, F.J., D. Przybylski, S. Yin, T. Sharpe, S. Gnerre, A. Abouelleil, et al. 2012. Finished bacterial genomes from shotgun sequence data. *Genome Research* 22(11):2270–2277.

Rimmer, A., H. Phan, I. Mathieson, Z. Iqbal, S.R.F. Twigg, WGS500 Consortium, A.O.M. Wilkie, G. McVean, and G. Lunter. 2014. Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nat Genet* 46(8):912–918.

Robertson, G., J. Schein, R. Chiu, R. Corbett, M. Field, S.D. Jackman, et al. 2010. *De novo* assembly and analysis of RNA-seq data. *Nat Meth* 7(11):909–912.

Robinson, J.T., H. Thorvaldsdóttir, W. Winckler, M. Guttman, E.S. Lander, G. Getz, and J.P. Mesirov. 2011. Integrative genomics viewer. *Nat Biotech* 29(1):24–26.

Roche Diagnostics Corporation. 1996. 454 Sequencing. `http://454.com`. Accessed: 2016-06-03.

Rodríguez-Ezpeleta, N., M. Hackenberg, and A.M. Aransay. 2012. *Bioinformatics for High Throughput Sequencing*. Springer.

Ronaghi, M., S. Karamohamed, B. Pettersson, M. Uhlén, and P. Nyrén. 1996. Real-time DNA sequencing using detection of pyrophosphate release. *Analytical Biochemistry* 242(1):84–89.

Ronaghi, M., M. Uhlén, and P. Nyrén. 1998. A sequencing method based on real-time pyrophosphate. *Science* 281(5375):363–365.

Rothberg, J.M., W. Hinz, T.M. Rearick, J. Schultz, W. Mileski, et al. 2011. An integrated semiconductor device enabling non-optical genome sequencing. *Nature* 475(7356):348–352.

Rothberg, J.M., and J.H. Leamon. 2008. The development and impact of 454 sequencing. *Nat Biotech* 26(10):1117–1124.

Ruffalo, M., T. LaFramboise, and M. Koyutürk. 2011. Comparative analysis of algorithms for next-generation sequencing read alignment. *Bioinformatics* 27(20):2790–2796.

Ruffalo, Matthew, M. Koyutürk, S. Ray, and T. LaFramboise. 2012. Accurate estimation of short read mapping quality for next-generation genome sequencing. *Bioinformatics* 28(18):i349–i355.

Rumble, S.M., P. Lacroute, A.V. Dalca, M. Fiume, A. Sidow, and M. Brudno. 2009. SHRiMP: Accurate mapping of short color-space reads. *PLoS Comput Biol* 5(5):1–11.

Rusk, N., and V. Kiermer. 2008. Primer: Sequencing – the next generation. *Nat Meth* 5(1):15–15.

Salzberg, S.L., and J.A. Yorke. 2005. Beware of mis-assembled genomes. *Bioinformatics* 21(24):4320–4321.

SAMtools Project. 2015. Hts-specs – VCFv4.3. `http://samtools.github.io/hts-specs/VCFv4.3.pdf`. Accessed: 2016-06-03.

Sanger, F., S. Nicklen, and A.R. Coulson. 1977. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America* 74(12):5463–5467.

Schadt, Eric E., S. Turner, and A. Kasarskis. 2010. A window into third-generation sequencing. *Human Molecular Genetics* 19(R2):R227–R240.

Schatz, M.C., A.L. Delcher, and S.L. Salzberg. 2010. Assembly of large genomes using second-generation sequencing. *Genome Research* 20(9):1165–1173.

Schatz, M.C., J. Witkowski, and W.R. McCombie. 2012. Current challenges in *de novo* plant genome sequencing and assembly. *Genome Biology* 13(4):243–243.

Scherer, S.W., C. Lee, E. Birney, D.M. Altshuler, E.E. Eichler, N.P. Carter, M.E. Hurles, and L. Feuk. 2007. Challenges and standards in integrating surveys of structural variation. *Nat Genet* 39(7 Suppl):S7–15.

Schulz, M.H., D.R. Zerbino, M. Vingron, and E. Birney. 2012. Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels. *Bioinformatics* 28(8):1086–1092.

Schuster, S.C. 2008. Next-generation sequencing transforms today's biology. *Nat Meth* 5(1):16–18.

Shang, J., F. Zhu, W. Vongsangnak, Y. Tang, W. Zhang, and B. Shen. 2014. Evaluation and comparison of multiple aligners for Next-Generation Sequencing data analysis. *BioMed Research International* 2014:16.

Shao, W., V.F Boltz, J.E. Spindler, M.F. Kearney, F. Maldarelli, J.W. Mellors, et al. 2013. Analysis of 454 sequencing error rate, error sources, and artifact recombination for detection of Low-frequency drug resistance mutations in HIV-1 DNA. *Retrovirology* 10(1):1–16.

Shendure, J., and H. Ji. 2008. Next-generation DNA sequencing. *Nat Biotech* 26(10):1135–1145.

Shendure, J., G.J. Porreca, N.B. Reppas, X. Lin, J.P. McCutcheon, A.M. Rosenbaum, M.D. Wang, K. Zhang, R.D. Mitra, and G.M. Church. 2005. Accurate multiplex polony sequencing of an evolved bacterial genome. *Science* 309(5741):1728–1732.

Sherry, S.T., M.H. Ward, M. Kholodov, J. Baker, L. Phan, E.M. Smigielski, and K. Sirotkin. 2001. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research* 29(1):308–311.

Simola, D., and J. Kim. 2011. Sniper: improved SNP discovery by multiply mapping deep sequenced reads. *Genome Biology* 12(6):R55.

Simpson, J.T., and R. Durbin. 2012. Efficient *de novo* assembly of large genomes using compressed data structures. *Genome Research* 22(3):549–556.

Simpson, J.T., K. Wong, S.D. Jackman, J.E. Schein, S.J. Jones, and I. Birol. 2009. ABySs: a parallel assembler for short read sequence data. *Genome Research* 19(6):1117–23.

Sims, D., I. Sudbery, N.E. Ilott, A. Heger, and C.P. Ponting. 2014. Sequencing depth and coverage: key considerations in genomic analyses. *Nat Rev Genet* 15(2):121–132.

Smith, A.D., Z. Xuan, and M.Q. Zhang. 2008. Using quality scores and longer reads improves accuracy of Solexa read mapping. *BMC Bioinformatics* 9(1): 1–8.

Smith, L.M., J.Z. Sanders, R.J. Kaiser, P. Hughes, C. Dodd, C.R. Connell, C. Heiner, S.B.H. Kent, and L.E. Hood. 1986. Fluorescence detection in automated DNA sequence analysis. *Nature* 321(6071):674–679.

Smolka, M., P. Rescheneder, M. C. Schatz, Arndt von Haeseler, and F.J. Sedlazeck. 2015. Teaser: Individualized benchmarking and optimization of read mapping results for NGS data. *Genome Biology* 16(1):235.

SNPedia. 2008. Glossary. `http://snpedia.com/index.php/Glossary`. Accessed: 2016-03-03.

Snyder, M.W., A. Adey, J.O. Kitzman, and J. Shendure. 2015. Haplotype-resolved genome sequencing: experimental methods and applications. *Nat Rev Genet* 16(6):344–358.

SPBAU. 2013. QUAST 2.1 manual. `http://quast.bioinf.spbau.ru/manual.html`. Accessed: 2014-05-29.

St. John, John. 2014. jstjohn/simseq. `https://github.com/jstjohn/SimSeq`. Accessed: 2014-11-29.

Staden, R. 1979. A strategy of DNA sequencing employing computer programs. *Nucleic Acids Research* 6(7):2601–2610.

Sultan, M., M.H. Schulz, H. Richard, A. Magen, A. Klingenhoff, M. Scherf, et al. 2008. A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science* 321(5891):956–960.

Swerdlow, H., and R. Gesteland. 1990. Capillary gel electrophoresis for rapid, high resolution DNA sequencing. *Nucleic Acids Research* 18(6):1415–1419.

Talwalkar, A., J. Liptrap, J. Newcomb, C. Hartl, J. Terhorst, K. Curtis, et al. 2014. SMaSH: a benchmarking toolkit for human genome variant calling. *Bioinformatics* 30(19):2787–2795.

Taub, M.A., H.C. Bravo, and R.A. Irizarry. 2010. Overcoming bias and systematic errors in next generation sequencing data. *Genome Medicine* 2(12):1–5.

Taylor, K.H., R.S. Kramer, J.W. Davis, J. Guo, D.J. Duff, D. Xu, C.W. Caldwell, and H. Shi. 2007. Ultradeep bisulfite sequencing analysis of DNA methylation patterns in multiple gene promoters by 454 sequencing. *Cancer Research* 67(18): 8511–8518.

The 1000 Genomes Project Consortium. 2010. A map of human genome variation from population scale sequencing. *Nature* 467(7319):1061–1073.

The Arabidopsis Genome Initiative. 2000. Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature* 408(6814):796–815.

The Arabidopsis Information Resource. 2010. Tair10_intergenic_20101028. `ftp://ftp.arabidopsis.org/home/tair/Sequences/blast_datasets/TAIR10_blastsets/TAIR10_intergenic_20101028`. Accessed: 2013-10-15.

The Arabidopsis Information Resource. 2011a. Index of /home/tair/Sequences/whole_chromosomes. `ftp://ftp.arabidopsis.org/home/tair/Sequences/whole_chromosomes`. Accessed: 2013-10-15.

The Arabidopsis Information Resource. 2011b. Tair10_seq_20110103_representative_gene_model_updated. `ftp://ftp.arabidopsis.org/home/tair/Sequences/blast_datasets/TAIR10_blastsets/TAIR10_seq_20110103_representative_gene_model_updated`. Accessed: 2013-10-15.

The Genome Factory. 2012. Using Velvet with mate-pair sequences. `http://thegenomefactory.blogspot.co.uk/2012/09/using-velvet-with-mate-pair-sequences.html`. Accessed: 2014-11-20.

The Molecular Ecologist. 2014. 2014 NGS Field Guide: Overview. `http://www.molecularecologist.com/next-gen-fieldguide-2014/`. Accessed: 2014-10-18.

Thermo Fisher Scientific, Inc. 2015. Ion Torrent. https://www.thermofisher.com/uk/en/home/brands/ion-torrent.html. Accessed: 2016-06-03.

Thompson, J.F., and P.M. Milos. 2011. The properties and applications of single-molecule DNA sequencing. *Genome Biology* 12(2):1–10.

Thudi, M., Y. Li, S.A. Jackson, G.D. May, and R.K. Varshney. 2012. Current state-of-art of sequencing technologies for plant genomics research. *Briefings in Functional Genomics* 11(1):3–11.

Trapnell, C., and S.L. Salzberg. 2009. How to map billions of short reads onto genomes. *Nat Biotech* 27(5):455–457.

Treangen, T.J., and S.L. Salzberg. 2012. Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nat Rev Genet* 13(1): 36–46.

UC Davis Genome Center. 2011. The Assemblathon. http://assemblathon.org. Accessed: 2015-05-29.

USDEOS, United States Department of Energy Office of Science. 2008. Genomics and its impact on science and society. Genomics and Its Impact on Science and Society.

Valouev, A., J. Ichikawa, T. Tonthat, J. Stuart, S. Ranade, H. Peckham, K. Zeng, J.A. Malek, G. Costa, K. McKernan, A. Sidow, A. Fire, and S.M. Johnson. 2008. A high-resolution, nucleosome position map of *C. elegans* reveals a lack of universal sequence-dictated positioning. *Genome Research* 18(7):1051–1063.

Victoria Wang, X., N. Blades, J. Ding, R. Sultana, and G. Parmigiani. 2012. Estimation of sequencing error rates in short reads. *BMC Bioinformatics* 13(1): 1–12.

Victorian Bioinformatics Consortium. 2012. VelvetOptimiser. http://bioinformatics.net.au/software.velvetoptimiser.shtml. Accessed: 2013-10-15.

Vignal, A., D. Milan, M. SanCristobal, and A. Eggen. 2002. A review on SNP and other types of molecular markers and their use in animal genetics. *Genetics, Selection, Evolution : GSE* 34(3):275–305.

Wang, J., Wei Wang, Ruiqiang Li, Yingrui Li, Geng Tian, Laurie Goodman, et al. 2008. The diploid genome sequence of an Asian individual. *Nature* 456(7218): 60–65.

Wang, K., M. Li, and H. Hakonarson. 2010. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research* 38(16):e164–e164.

Warren, R.L., G.G. Sutton, S.J.M. Jones, and R.A. Holt. 2007. Assembling millions of short DNA sequences using SSAKE. *Bioinformatics* 23(4):500–501.

Waterston, R.H., E.S. Lander, and J.E. Sulston. 2002. On the sequencing of the human genome. *Proceedings of the National Academy of Sciences* 99(6): 3712–3716.

Weber, A.P.M. 2015. Discovering new biology through RNA-Seq. *Plant Physiology* 169(3):1524–31.

Whiteford, N., T. Skelly, C. Curtis, M.E. Ritchie, A. Löhr, A.W. Zaranek, I. Abnizova, and C. Brown. 2009. Swift: primary data analysis for the illumina solexa sequencing platform. *Bioinformatics* 25(17):2194–2199.

Wikipedia. 2005. Sequence assembly. `https://en.wikipedia.org/wiki/Sequence_assembly`. Accessed: 2016-04-24.

Wikipedia. 2008. List of Sequence Alignment Software. `https://en.wikipedia.org/wiki/List_of_sequence_alignment_software`. Accessed: 2016-04-25.

Wikipedia. 2014. SNV calling from NGS data. `https://en.wikipedia.org/wiki/SNV_calling_from_NGS_data`. Accessed: 2016-04-24.

Wold, B., and R.M. Myers. 2008. Sequence census methods for functional genomics. *Nat Meth* 5(1):19–21.

Wu, H., R.A. Irizarry, and H.C. Bravo. 2010. Intensity normalization improves color calling in SOLiD sequencing. *Nat Meth* 7(5):336–337.

Yandell, M., and D. Ence. 2012. A beginner's guide to eukaryotic genome annotation. *Nat Rev Genet* 13(5):329–342.

Yi, X., Y. Liang, E. Huerta-Sanchez, X. Jin, Z.X. Cuo, J.E. Pool, X. Xu, et al. 2010. Sequencing of 50 human exomes reveals adaptation to high altitude. *Science* 329(5987):75–78.

You, N., G. Murillo, X. Su, X. Zeng, J. Xu, K. Ning, S. Zhang, J. Zhu, and X. Cui. 2012. SNP calling using genotype model selection on high-throughput sequencing data. *Bioinformatics* 28(5):643–650.

yourgenome.org. 2015. What is the 454 method of DNA sequencing? `http://www.yourgenome.org/facts/what-is-the-454-method-of-dna-sequencing`. Accessed: 2016-06-03.

Yu, X., K. Guda, J. Willis, M. Veigl, Z. Wang, S. Markowitz, M.D. Adams, and S. Sun. 2012. How do alignment programs perform on sequencing data with varying qualities and from repetitive regions? *BioData Mining* 5:6–6.

Yu, X., and S. Sun. 2013. Comparing a few SNP calling algorithms using low-coverage sequencing data. *BMC Bioinformatics* 14(1):1–15.

Zerbino, D.R. 2010. Using the Velvet *de novo* assembler for short-read sequencing technologies. *Curr Protoc Bioinformatics* Chapter 11. Unit–11.5.

Zerbino, D.R., and E. Birney. 2008. Velvet: Algorithms for *de novo* short read assembly using de Bruijn graphs. *Genome Res* 18(5):821–829.

Zerbino, D.R., G.K. McEwen, E.H. Margulies, and E. Birney. 2009. Pebble and Rock Band: Heuristic resolution of repeats and scaffolding in the Velvet short-read *de Novo* assembler. *PLoS ONE* 4(12):e8407.

Zhang, J., R. Chiodini, A. Badr, and G. Zhang. 2011. The impact of next-generation sequencing on genomics. *Journal of genetics and genomics* 38(3):95–109.

Zimin, A.V., G. Marçais, D. Puiu, M. Roberts, S.L. Salzberg, and J.A. Yorke. 2013. The MaSuRCA genome assembler. *Bioinformatics* 29(21):2669–2677.

Zook, J., and M. Salit. 2011. Genomes in a bottle: creating standard reference materials for genomic variation – why, what and how? *Genome Biology* 12(Suppl 1):P31.

Appendices

Appendix A

A.1 False positive SNP generation due to reference misassembly – supplementary information

A.1.1 Commands, parameters, and some detailed results of the experiment with real RNA-Seq data from barley cultivar Bowman to test for reference misassembly

A.1.1.1 Barley cultivar Bowman transcriptome *de novo* assembly basic statistics

- Output of the TrinityStats.pl script (version trinityrnaseq_r20140717):

```
################################
## Counts of transcripts, etc.
################################
Total trinity 'genes':  40836
Total trinity transcripts:  53336
Percent GC: 49.41

########################################
Stats based on ALL transcript contigs:
########################################

 Contig N10:  3164
 Contig N20:  2377
 Contig N30:  1942
```

```
 Contig N40:  1604
 Contig N50:  1311



 Median contig length:  475
 Average contig:  804.53
 Total assembled bases:  42910338


#####################################################
## Stats based on ONLY LONGEST ISOFORM per 'GENE':
#####################################################

 Contig N10:  3114
 Contig N20:  2298
 Contig N30:  1858
 Contig N40:  1512
 Contig N50:  1209

 Median contig length:  394
 Average contig:  717.55
 Total assembled bases:  29301963
```

A.1.1.2 Pipeline usage in the experiment with real RNA-Seq data from barley cultivar Bowman

Command usage: `java fps.AnalyzeHomozygousSNPs <FASTA file> <Truncate name at space character? true | false> <List of Homozygous SNPs file> <Absolute path to Trinity .reads directory> <Number of mismatches for BWA alignment> <Number of BWA running threads> <outputFile> <Run BLAST against any specific database? true | false> <Target BLAST database file | none>`

where:

`<FASTA file>` The absolute path of the `Trinity.fasta` file with the barley cultivar Bowman transcriptome assembly.

`<Truncate name at space character? true | false>`

`true` should be chosen if the transcripts' names have spaces.

`<List of Homozygous SNPs file>`

The absolute path of the text file with the list of homozygous SNPs to be analysed by the pipeline.

`<Absolute path of Trinity .reads directory>`

The absolute path of the directory where the .reads files of Trinity reside.

`<Number of mismatches for BWA alignment>`

The number of allowed mismatches for the BWA relaxed run.

`<Number of BWA running threads>`

The number of parallel threads assigned for BWA to run.

`<outputFile>` The name to be assigned for the output file.

`<Run BLAST against any specific database? true | false>`

`true` should be chosen when running the tool in paralogy test mode.

`<Target BLAST database file | none>`

When running the tool in the paralogy test mode, the absolute

path of the BLAST database file. <none> should be chosen when the previous parameter is set to <false>.

A.1.2 Commands, parameters, and some detailed results of the *de novo* assembly experiment with simulated reads from *Arabidopsis thaliana* to test for reference misassembly

A.1.2.1 Sherman parameters used in the read simulation stage

Command usage: `Sherman --length <length of sequences to be generated> -n <number of sequences to be generated> --genome_folder <path to chromosome folder> -pe -cr 0 -e 0`

`--length` <length of sequences to be generated>

In this case, 150.

`-n` <number of sequences to be generated>

Calculated based on factors such as the aimed coverage depth, read sequence length, and chromosome sequence length. Table A.1 shows the values set for this experiment based on each chromosome length and the aimed 100-fold coverage.

`--min-frag` <the minimum size for paired-end fragments>

Table A.1: Number of 150 bp paired-end reads generated with Sherman tool aiming 100-fold coverage for each chromosome of *A. thaliana*.

Chromosome	Length (bp)	# 150 bp PE reads generated
Chr1	30,427,671	10,142,557
Chr2	19,698,289	6,566,096
Chr3	23,459,830	7,819,943
Chr4	18,585,056	6,195,019
Chr5	26,975,502	8,991,834

Abbreviations: Chr: Chromosome; bp: base pairs; #: Number of; PE: paired-end

Not explicitly set in the experiment to use the default value.

`-X` `<the maximum size for paired-end fragments>`

Not explicitly set in the experiment to use the default value.

`--genome_folder` `<path to chromosome folder>`

`-pe` for paired-end read files creation.

`-cr 0` to simulate standard genomic sequences.

`-e 0` for no introduction of sequencing errors.

Additional information can be found at Babraham Bioinformatics (2013a).

A.1.2.2 Velvet parameters used in the *de novo* assembly stage

- velveth step:

Command usage: `velveth directory hash_length [-file_format][-read_type]`

`[-separate|-interleaved] filename1 [filename2 ...]` {...} `[options]`

Usage example: `velveth . 99 -shortPaired -fastq TAIR10_all5Chrms_simul_`

`150bp_pe_shuffled.fastq`

where:

`directory`	Directory name for placing the output files.
`-hash_length`	Odd integer (if even, it will be decremented) <= 99 (if above, will be reduced). In this case, set to 99.
`[-file_format]`	File format option. In this case, `fastq`.
`[-read_type]`	Read type option. In this case, `-shortPaired`.
`filename1`	Shuffled FASTQ reads file originated by the accessory script `shuffleSequences_fastq.pl` after its run over the original paired-end reads generated by the read simulator.

- velvetg step:

Command usage: `velvetg directory [options]`

Usage example: `velvetg . -read_trkg yes -amos_file yes -exp_cov auto -cov_cutoff auto`

where:

`directory`	Working directory name.

`-read_trkg <yes|no>`	Tracking of short read positions in assembly. In this case, `yes`.
`-amos_file <yes|no>`	Export assembly to AMOS file. In this case, `yes`.
`-exp_cov <floating point|auto>`	Expected coverage of unique regions or allow the system to infer it. In this case, `auto`.
`-cov_cutoff <floating point|auto>`	Removal of low coverage nodes AFTER tour bus or allow the system to infer it. In this case, `auto`.

Additional information can be found in Velvet version 1.1 manual at EMBL-EBI (2008).

A.1.2.3 Velvet assembly QUAST results

Table A.2 shows the QUAST results for the *A. thaliana* Velvet assembly.

Table A.2: QUAST results for the *A. thaliana* Velvet assembly.

Assembly	*A. thaliana* Velvet assembly
# contigs (>= 0 bp)	7042
# contigs (>= 1000 bp)	1903
Total length (>= 0 bp)	117324215
Total length (>= 1000 bp)	115533308
# contigs	2795
Largest contig	1181083
Total length	116147737
Reference length	119146348
GC (%)	35.99
Reference GC (%)	36.03
N50	216151
NG50	210850
N75	100755
NG75	92700
L50	149
LG50	156
L75	349
LG75	372
# misassemblies	53
# misassembled contigs	47
Misassembled contigs length	5384293
# local misassemblies	792
# unaligned contigs	3 + 9 part
Unaligned length	9039
Genome fraction (%)	97.375
Duplication ratio	1.001
# N's per 100 kbp	73.30
# mismatches per 100 kbp	1.80
# indels per 100 kbp	1.39
# genes	27776 + 501 part
Largest alignment	1181083
NA50	211392
NGA50	207568
NA75	97744
NGA75	89971

Continued on next page

Table A.2 – continued from previous page

Assembly	*A. thaliana* Velvet assembly
LA50	154
LGA50	161
LA75	359
LGA75	383

Abbreviations: bp: base pairs; #: Number of

A.1.2.4 Bowtie2 STRICT mapping parameters

Command usage: `bowtie2 [options]* -x <bt2-idx>`

`{-1 <m1> -2 <m2> | -U <r>} -S [<hit>]`

where:

Usage example: `bowtie2 -x VelvetAssy_Bowtie2Index`

`-1 TAIR10_all5Chrms_simul_150bp_1.fastq -2 TAIR10_all5Chrms_simul_`

`150bp_2.fastq -S VelvetAssy_Bowtie2_aligned_cf-0.12.sam --phred33`

`--score-min L,0,-0.12 -p 8 --un-gz VelvetAssy_Bowtie2_`

`unal_cf-0.12.sam.gz --al-gz VelvetAssy_Bowtie2_once_cf-0.12.sam.gz`

`--un-conc-gz VelvetAssy_Bowtie2_unconc_cf-0.12.sam.gz --al-conc-gz`

`VelvetAssy_Bowtie2_conc_cf-0.12.sam.gz --rg-id VelvetAssy_Bowtie2_1mm`

`--rg SM:Pool1`

`-x <bt2-idx>`	The basename of the index for the reference genome.
`-1 <m1>`	Comma-separated list of files containing mate 1s.

`-1 <m2>`	Comma-separated list of files containing mate 2s.
`-S <hit>`	File to write SAM alignments to.
`--phred33`	Input qualities are ASCII characters equal to the Phred quality plus 33. This is also called the "Phred+33" encoding, which is used by the very latest Illumina pipelines.
`--score-min <function>`	Sets a function governing the minimum alignment score needed for an alignment to be considered "valid" (i.e. good enough to report). This is a function of read length. For instance, specifying `L,0,-0.12` sets the minimum-score `function f to f(x) = 0 + -0.12 * x`, where `x` is the read length. In this case, -0.12 (coefficient) * 150 bp = -18 will be equivalent to 3 times the default mismatch penalty of -6 or, in other words, 3 mismatches per 150 bp of read length. `L` means a linear function. `0` is the constant term of the function.
`-p NTHREADS`	Launch NTHREADS parallel search threads. In this case, 8 threads assigned.
`--un-gz <path>`	Write unpaired reads that fail to align to file at <path>. These reads correspond to the SAM records with the FLAGS `0x4` bit set and neither the `0x40` nor `0x80` bits set. If `--un-gz` is specified, output will be gzip compressed.
`--al-gz <path>`	Write unpaired reads that align at least once to file at <path>.

	These reads correspond to the SAM records with the FLAGS `0x4`, `0x40`, and `0x80` bits set.
	If `--al-gz` is specified, output will be gzip compressed.
`--un-conc-gz <path>`	Write paired-end reads that fail to align concordantly to file(s) at <path>. These reads correspond to the SAM records with the FLAGS `0x4` bit set and either the `0x40` or `0x80` bit set (depending on whether it's mate #1 or #2). If `--un-conc-gz` is specified, output will be gzip compressed.
`--al-conc-gz <path>`	Write paired-end reads that align concordantly at least once to file(s) at <path>. These reads correspond to the SAM records with the FLAGS `0x4` bit unset and either the `0x40` or `0x80` bit set (depending on whether it's mate #1 or #2). If `--al-conc-gz` is specified, output will be gzip compressed.
`--rg-id <text>`	Set the read group ID to <text>. In this case, `VelvetAssy_Bowtie2_1mm`.
`--rg <text>`	Add <text> (usually of the form `TAG:VAL`, e.g. `SM:Pool1`) as a field on the `@RG` header line.

Additional information can be found in Bowtie2 manual section *Setting function options* at Johns Hopkins University (2014).

After the generation of the SAM file, this was converted to the BAM (binary) format, sorted, and indexed, for the subsequent downstream analysis, with SAMtools package version 0.1.18.

A.1.2.5 Bowtie2 STRICT mapping results

Output from Bowtie2:

```
39715449 reads; of these:
  39715449 (100.00%) were paired; of these:
    10315746 (25.97%) aligned concordantly 0 times
    29018330 (73.07%) aligned concordantly exactly 1 time
    381373 (0.96%) aligned concordantly >1 times
    ----
    10315746 pairs aligned concordantly 0 times; of these:
      9374436 (90.88%) aligned discordantly 1 time
    ----
    941310 pairs aligned 0 times concordantly or discordantly; of these:
      1882620 mates make up the pairs; of these:
        471364 (25.04%) aligned 0 times
        501588 (26.64%) aligned exactly 1 time
        909668 (48.32%) aligned >1 times
99.41% overall alignment rate
```

A.1.2.6 FreeBayes parameters used when calling SNPs over the Bowtie2 STRICT mapping

Command usage: `freebayes [OPTION] ... [BAM FILE] ...`

Usage example: `freebayes -b VelvetAssy_Bowtie2_aligned_cf-0.12.sorted.bam -f VelvetAssy.fa -v VelvetAssy_Bowtie2_aligned_cf-0.12.vcf --haplotype-length 0 --min-alternate-count 4 --min-alternate-fraction 0`

```
--min-alternate-total 4 --pooled-continuous --ploidy 1
```

where:

`-b <in.bam>`	Add `FILE` to the set of BAM files to be analysed.
`-f <in.reference>`	Use `FILE` as the reference sequence for analysis.
`-v <out.vcf>`	Output VCF-format results to `FILE`.
`--haplotype-length N`	Allow haplotype calls with contiguous embedded matches of up to this length. (default: 3). In this case, set as `0` to perform naiver variant calling aiming simply annotation of observation counts of SNPs and indels.
`--min-alternate-count N`	Require at least this count of observations supporting an alternate allele within the total population in order to use the allele in analysis. (default: 1). In this case, set as `4`.
`--min-alternate-fraction N`	Require at least this fraction of observations supporting an alternate allele within a single individual in order to evaluate the position. (default: 0.2). In this case, set as `0` to perform naiver variant calling aiming simply annotation of observation counts of SNPs and indels.
`--min-alternate-total N`	Require at least this count of observations supporting an alternate allele within the total population in order to use the allele in analysis. (default: 1). In this case, set as `4`.

`--pooled-continuous` Output all alleles which pass input filters, regardless of genotyping outcome or model. Set to perform naiver variant calling aiming simply annotation of observation counts of SNPs and indels.

`--ploidy N` Sets the default ploidy for the analysis to N. (default: 2). In this case, set as `1`.

Additional information can be found at Garrison (2012) and in the FreeBayes embedded `--help` option.

A.1.2.7 Pipeline usage in the *de novo* assembly experiment with simulated reads from *Arabidopsis thaliana*

Command usage: `java fps.AnalyzeHomsSNPs <FASTA file> <Truncate name at space character? true | false> <List of Homozygous SNPs file> <Path to the assembly directory> <AFG file produced by Velvet> <Sequences file produced by Velvet> <Number of mismatches for Bowtie2 mapping> <Number of Bowtie2 running threads> <outputFile> <Run BLAST against any specific database? true | false> <Target BLAST database file | none>`

where:

`<FASTA file>` The absolute path of the Velvet assembly file.

`<Truncate name at space character? true | false>`

`true` should be chosen if the contigs' names have spaces.

`<List of Homozygous SNPs file>`

The absolute path of the text file with the list of homozygous SNPs to be analysed by the pipeline.

`<Path to the assembly directory>`

The absolute path of the Velvet assembly directory.

`<AFG file produced by Velvet>`

The absolute path of the Velvet assembly directory.

`<Sequences file produced by Velvet>`

The absolute path of the Velvet assembly directory.

`<Number of mismatches for Bowtie2 mapping>`

The number of allowed mismatches for the Bowtie2 relaxed runs. In this case, values chosen for each specific run were 5, 10, 20, and 30.

`<Number of Bowtie2 running threads>`

The number of parallel threads assigned for Bowtie2 to run.

`<outputFile>` The name to be assigned for the output file.

`<Run BLAST against any specific database? true | false>`

`true` should be chosen when running the tool in paralogy test mode.

<Target BLAST database file | none>

When running the tool in the paralogy test mode, the absolute path of the BLAST database file. <none> should be chosen when the previous parameter is set to <false>.

A.1.2.8 Bowtie2 RELAXED mapping parameters

These were internally handled by the pipeline for each SNP event per assembled contig but, basically, were analogous to the parameters detailed above in item A.1.2.4. The only major difference was regarding the values applied in the function --score-min <func> depending on the relaxed mismatch setting. These were set accordingly to the Table A.3 shown below.

Table A.3: Values set for the --score-min <func> depending on the aimed relaxed mismatch setting.

Number of mismatches	--score-min <func> function
5	--score-min L,0,-0.2
10	--score-min L,0,-0.4
20	--score-min L,0,-0.8
30	--score-min L,0,-1.2

A.1.2.9 Bowtie2 RELAXED mapping results

These were internally handled by the developed pipeline, for each SNP event per assembled contig, with the SAMtools 'flagstat' command. A summary of the average percentage

of mapped reads for each relaxed mismatch setting is shown in Table A.4 below.

Table A.4: Average percentage of mapped reads for the SNP events analysed by the pipeline in each relaxed mismatch setting.

Number of mismatches	Average percentage of mapped reads (%)
5	80.83
10	85.75
20	94.08
30	99.82

A.1.2.10 Most relaxed mapping paralog test results

Table A.5: *A. thaliana* annotation BLASTDB hits found by the pipeline for cases with either the reference or the alternate alleles present in the reads sets considering the most relaxed mapping scenario.

contig	length	SNP pos.	ref. allele	ref. allele hit(s)	alt. allele(s) hit(s) / (allele)
NODE_6286	201	172	T	AT3G30680.1 - t.e.g. - chr3:12228316-12232218; AT1G35590.1 - t.e.g. - chr1:13131935-13135837	AT4G03900.1 - t.e.g. - chr4:1838791-1842694 / (A)
NODE_18482	1,044	1,032	C	AT5G33234.1 - t.e.g. - chr5:12483583-12487437	AT5G59640.1 - t.e.g. - chr5:24025992-24030772 / (T)
NODE_1136	344	304	G	AT5G36870.1 - glucan synthase-like 9 - chr5:14518316-14533930	AT2G15310-AT2G15318 - int. - chr2:6655502-6664759 / (A)
NODE_12708	197	26	C	AT3G44235-AT3G44240 - int. - chr3:15933964-15940183; AT4G02280-AT4G02290 - int. - chr4:998968-1002393	AT3G48630-AT3G48640 - int. - chr3:18018809-18021649 / (T)
NODE_12708	197	31	C	AT3G44235-AT3G44240 - int. - chr3:15933964-15940183; AT4G02280-AT4G02290 - int. - chr4:998968-1002393	AT3G48630-AT3G48640 - int. - chr3:18018809-18021649 / (T)

Continued on next page

contig	length	SNP pos.	ref. allele	ref. allele hit(s)	alt. allele(s) hit(s) / (allele)
NODE_9420	240	27	C	AT4G08092.1 - t.e.g. - chr4:4984426-4988683; AT5G28165.1 - t.e.g. - chr5:10140418-10147542	AT5G29056.1 - t.e.g. - chr5:11117609-11121412 / (T); AT4G08000.1 - t.e.g. - chr4:4831118-4833406 / (T)
NODE_9420	240	45	C	AT4G08092.1 - t.e.g. - chr4:4984426-4988683; AT5G28165.1 - t.e.g. - chr5:10140418-10147542	AT5G29056.1 - t.e.g. - chr5:11117609-11121412 / (T); AT4G08000.1 - t.e.g. - chr4:4831118-4833406 / (T)
NODE_3278	308,548	48	C	AT2G46710-AT2G46720 - int. - chr2:19194850-19197382	AT1G07270-AT1G07280 - int. - chr1:2232898-2238086 / (T)
NODE_7782	504,912	504,883	C	AT4G29080-AT4G29090 - int. - chr4:14325329-14333527; AT4G10760-AT4G10767 - int. - chr4:6623352-6627035	AT3G20030-AT3G20040 - int. - chr3:6991463-6994893 / (T)

Continued on next page

contig	length	SNP pos.	ref. allele	ref. allele hit(s)	alt. allele(s) hit(s) / (allele)
NODE_3266	50,156	49,470	T	AT2G29995-AT2G30000 - int. - chr2:12799420-12803852	AT2G28315-AT2G28320 - int. - chr2:12090702-12094745 / (C); AT3G27831-AT3G27835 - int. - chr3:10320093-10321740 / (C); AT1G34010-AT1G34020 - int. - chr1:12361190-12366853 / (C); AT2G15310-AT2G15318 - int. - chr2:6655502-6664759 / (C); AT1G28120-AT1G28130 - int. - chr1:9815545-9825285 / (C); AT4G35510-AT4G35519 - int. - chr4:16862373-16865247 / (C); AT2G02135-AT2G02140 - int. - chr2:542174-544706 / (C); AT1G43270-AT1G43280 - int. - chr1:16324363-16330758 / (C); AT3G28800-AT3G28810 - int. - chr3:10818114-10822470 / (C); AT3G22555.1 - t.e.g. - pseudogene, putative DNA methyltransferase chr3:7995486-7996636 / (C); AT5G46490-AT5G46500 - int. - chr5:18853844-18856453 / (C); AT1G52270-AT1G52280 - int. - chr1:19464356-19467968 / (C); AT3G46380-AT3G46382 - int. - chr3:17059897-17063424 / (C); AT5G13450-AT5G13460 - int. - chr5:4312049-4315758 / (C)

Continued on next page

contig	length	SNP pos.	ref. allele	ref. allele hit(s)	alt. allele(s) hit(s) / (allele)
NODE_10554	528	10	T	—	AT4G04394.1 - t.e.g. - chr4:2157650-2158105 / (C); AT3G33235-AT3G35003 - int. - chr3:14105291-14105721 / (C); AT2G14330-AT2G14335 - int. - chr2:6075395-6077329 / (C); AT3G34299-AT3G35707 - int. - chr3:14134677-14136756 / (C); AT4G09400.1 - t.e.g. - chr4:5953226-5958057 / (C)
NODE_13209	237	224	C	AT3G43350.1 - t.e.g. - chr3:15286073-15290808 (15289407-15290032 sub-region)	AT3G43350.1 - t.e.g. - chr3:15286073-15290808 (15289435-15289831 sub-region) / (G); AT3G43350.1 - t.e.g. - chr3:15286073-15290808 (15288578-15289190 sub-region) / (T)
NODE_5016	232	203	G	AT1G27110-AT1G27120 - int. - chr1:9417505-9421176; AT2G04047-AT2G04036 - int. - chr2:1317387-1327908; AT5G22550-AT5G22555 - int. - chr5:7485483-7489421; AT1G27110-AT1G27120 - int. - chr1:9417505-9421176	AT3G29755-AT3G29760 - int. - chr3:11585252-11589504 / (A)

Continued on next page

contig	length	SNP pos.	ref. allele	ref. allele hit(s)	alt. allele(s) hit(s) / (allele)
NODE_4138	237	6	C	AT3G30790.1 - t.e.g. - chr3:12467942-12471863; AT2G13000.1 - t.e.g. - chr2:5342940-5346917	AT5G19015.1 - t.e.g. - chr5:6349762-6350949 / (T)
NODE_1866	59,108	21	A	AT3G04945-AT3G04950 - int. - chr3:1369533-1371704; AT3G26134-AT3G26140 - int. - chr3:9557800-9559681; AT1G09520-AT1G09530 - int. - chr1:3072051-3076581; AT1G77990.1 - STAS domain - Sulfate transporter family chr1:29317899-29323352; AT1G62580.1 - Flavin-binding monooxygenase family protein - chr1:23173333-23176931	AT1G19450-AT1G19460 - int. - chr1:6734883-6738483 / (G); AT1G01030-AT1G01040 - int. - chr1:13715-23145 / (G)
NODE_11469	24,846	34	T	AT2G13160.1 - t.e.g. - chr2:5441880-5444055	AT4G02314.1 - t.e.g. - chr4:1018434-1020824 / (A)
NODE_2297	1,123	1,094	T	AT5G28526.1 - t.e.g. - chr5:10515532-10521998	AT4G08060.1 - t.e.g. - chr4:4924776-4928259 / (A)
NODE_2026	388	385	A	AT1G43110.1 - pseudogene, putative polygalacturonase (*Phleum pratense*) - chr1:16223981-16225810	AT1G43120.1 - pseudogene, putative polygalacturonase protein allergen (*Cynodon dactylon*) - chr1:16227318-16227779 / (T)

Continued on next page

contig	length	SNP pos.	ref. allele	ref. allele hit(s)	alt. allele(s) hit(s) / (allele)
NODE_11067	197	180	C	AT2G05860.1 - t.e.g. - chr2:2246959-2248108; AT5G35280.1 - t.e.g. - chr5:13509622-13510530	AT4G07310.1 - t.e.g. - chr4:4108477-4109489 / (T)
NODE_6507	209	202	T	AT3G42253.1 - t.e.g. - chr3:14411778-14413179; AT1G38185.1 - t.e.g. - chr1:14334140-14338177	AT3G43862.1 - t.e.g. - chr3:15714896-15716641 / (C); AT5G32053.1 - t.e.g. - chr5:11800200-11802224 / (C); AT5G32475.1 - t.e.g. - chr5:12102512-12104578 / (C)
NODE_3725	219	6	T	AT5G28526.1 - t.e.g. - chr5:10515532-10521998	AT2G14970.1 - t.e.g. - chr2:6457721-6461341 / (A); AT1G40109.1 - t.e.g. - chr1:15155045-15158202 / (A); AT4G08060.1 - t.e.g. - chr4:4924776-4928259 / (A); AT3G32950.1 - t.e.g. - chr3:13495365-13498632 / (A)
NODE_13084	92,365	92,356	G	AT3G27290-AT3G27300 - int. - chr3:10081534-10083048; AT3G13820-AT3G13825 - int. - chr3:4550864-4552523	AT5G15300-AT5G15310 - int. - chr5:4970035-4974670 / (C)
NODE_19912	249	207	C	AT5G35195-AT5G35200 - int. - chr5:13449750-13462161 (13457232-13457915 sub-region)	AT5G35195-AT5G35200 - int. - chr5:13449750-13462161 (13458965-13459646 sub-region) / (A)

Continued on next page

contig	length	SNP pos.	ref. allele	ref. allele hit(s)	alt. allele(s) hit(s) / (allele)
NODE_10891	230,391	43	G	AT4G15350-AT4G15360 - int. - chr4:8764595-8770184	AT2G36680-AT2G36690 - int. - chr2:15370847-15379693 / (A)
NODE_6148	308	33	G	AT1G23920.1 - t.e.g. - chr1:8454200-8456566; AT2G23720.1 - t.e.g. - chr2:10088774-10092878	AT4G09380.1 - t.e.g. - chr4:5946583-5950874 / (A); AT5G33232.1 - t.e.g. - chr5:12479274-12481736 / (A)
NODE_10293	235	56	C	AT5G23010-AT5G23020 - int. - chr5:7706897-7718120; AT2G14350-AT2G14365 - int. - chr2:6082162-6088338; AT2G29200-AT2G29210 - int. - chr2:12553434-12558050; AT3G16520-AT3G16530 - int. - chr3:5620874-5624376; AT2G13180-AT2G13190 - int. - chr2:5464638-5469692	AT5G48540-AT5G48543 - int. - chr5:19669921-19674391 / (T); AT4G11700-AT4G11710 - int. - chr4:7058638-7061148 / (T)
NODE_6185	267	233	A	AT1G32450-AT1G32460 - int. - chr1:11719971-11738117; AT5G35550-AT5G35555 - int. - chr5:13727861-13736244	AT3G29515-AT3G29520 - int. - chr3:11346096-11351104 / (T); AT1G43995-AT1G43997 - int. - chr1:16699057-16704095 / (T)

Continued on next page

contig	length	SNP pos.	ref. allele	ref. allele hit(s)	alt. allele(s) hit(s) / (allele)
NODE_6185	267	235	A	AT1G32450-AT1G32460 - int. - chr1:11719971-11738117; AT5G35550-AT5G35555 - int. - chr5:13727861-13736244	AT3G29515-AT3G29520 - int. - chr3:11346096-11351104 / (G); AT1G43995-AT1G43997 - int. - chr1:16699057-16704095 / (G)
NODE_637	445	441	T	AT1G40085-AT1G40087 - int. - chr1:14926971-14999614 (14945206-14945840 sub-region)	AT1G40085-AT1G40087 - int. - chr1:14926971-14999614 (14936590-14940266 sub-region) / (C)
NODE_12180	438	392	G	AT3G32393.1 - t.e.g. - chr3:13344240-13349880	AT3G43390.1 - t.e.g. - chr3:15326264-15330946 / (A)
NODE_13081	221	141	C	AT4G20730.1 - t.e.g. - chr4:11117798-11120595; AT5G14830.1 - t.e.g. - chr5:4794769-4798102; AT4G20500-AT4G20510 - int. - chr4:11041008-11041656	AT3G62490.1 - t.e.g. - chr3:23112500-23114999 / (T)

Continued on next page

contig	length	SNP pos.	ref. allele	ref. allele hit(s)	alt. allele(s) hit(s) / (allele)
NODE_6045	197	172	T	AT1G37150-AT1G37160 - int. - chr1:14177993-14181395	AT4G06736-AT4G06738 - int. - chr4:4043095-4044835 / (C); AT4G08040-AT4G08050 - int. - chr4:4888940-4895658 / (C); AT5G33389-AT5G33391 - int. - chr5:12647477-12649193 / (C); AT2G06250-AT2G06255 - int. - chr2:2450773-2457572 / (C)
NODE_6045	197	188	C	AT1G37150-AT1G37160 - int. - chr1:14177993-14181395	AT4G06736-AT4G06738 - int. - chr4:4043095-4044835 / (T); AT4G08040-AT4G08050 - int. - chr4:4888940-4895658 / (T); AT5G33389-AT5G33391 - int. - chr5:12647477-12649193 / (T); AT2G06250-AT2G06255 - int. - chr2:2450773-2457572 / (T)
NODE_5082	313	20	C	AT1G40107.1 - t.e.g. - chr1:15151216-15154824	AT5G28526.1 - t.e.g. - chr5:10515532-10521998 / (A); AT4G08070.1 - t.e.g. - chr4:4928374-4931801 / (A); AT2G14980.1 - t.e.g. - chr2:6461672-6465118 / (A)

Abbreviations: pos.: position; ref.: reference; alt.: alternate; t.e.g.: transposable element gene; int.: intergenic region; chr: chromosome

A.1.3 Software availability

The source code used in this specific study is available at:

https://github.com/acbellorib/fpSNPsProject.

Appendix B

B.1 False positive SNP generation due to read mismapping – supplementary information

B.1.1 Commands, parameters, and some detailed results of the *de novo* assembly experiment with simulated reads from *Arabidopsis thaliana* to test for read mismapping

B.1.1.1 Velvet parameters used in the *de novo* assembly stage

VelvetOptimiser.pl script parameters:

Command usage: `VelvetOptimiser.pl [options] -f 'velveth input line'`

Usage example:
```
VelvetOptimiser.pl --v --s 31 --e 99 --t 8
-f '-shortPaired -fastq TAIR10_all5Chrms_simul
_150bp_pe_shuffled.fastq'
```

where:

`--v`	Verbose logging; includes all velvet output in the logfile.
`--s`	The starting (lower) hash value (default '19').
`--e`	The end (higher) hash value (default '31').
`--t`	The maximum number of simultaneous velvet instances to run (default '4').
`-f 'velveth input line'`	In this case, set up with the options `fastq` as the type of file, `-shortPaired` as the read type, followed by the shuffled FASTQ reads file originated by the accessory script `shuffleSequences_fastq.pl` after its run over the original paired-end reads generated by the read simulator.

B.1.1.2 Velvet *de novo* assembly statistics

Table B.1 shows the output retrieved from the VelvetOptimiser.pl logging information.

Table B.2 shows the assembly obtained QUAST results.

Table B.1: Final optimised *A. thaliana* Velvet assembly details retrieved from the VelvetOptimiser.pl logging information.

Assembly	*A. thaliana* optimised Velvet assembly
Velvet hash value	99
Roadmap file size	5,017,355,025
Total number of contigs	7,043
n50	212,622
length of longest contig	1,181,083
Total bases in contigs	117,324,500
Number of contigs > 1k	1,903
Total bases in contigs > 1k	115,533,394
PE library length ; sample st. dev.	231 ; 95
PE library length ; sample st. dev.	232 ; 95

Abbreviations: PE: Paired-end; st. dev.: standard deviation

Table B.2: QUAST results for the *A. thaliana* optimised Velvet assembly.

Assembly	*A. thaliana* optimised Velvet assembly
# contigs (>= 0 bp)	7043
# contigs (>= 1000 bp)	1903
Total length (>= 0 bp)	117324500
Total length (>= 1000 bp)	115533394
# contigs	2795
Largest contig	1181083
Total length	116147823
Reference length	119146348
GC (%)	35.99
Reference GC (%)	36.03
N50	216151
NG50	210850
N75	100755
NG75	92700

Continued on next page

Table B.2 – continued from previous page

Assembly	*A. thaliana* optimised Velvet assembly
L50	149
LG50	156
L75	349
LG75	372
# misassemblies	53
# misassembled contigs	47
Misassembled contigs length	5384293
# local misassemblies	793
# unaligned contigs	3 + 9 part
Unaligned length	9039
Genome fraction (%)	97.375
Duplication ratio	1.001
# N's per 100 kbp	73.25
# mismatches per 100 kbp	1.80
# indels per 100 kbp	1.39
# genes	27775 + 502 part
Largest alignment	1181083
NA50	211392
NGA50	207568
NA75	97744
NGA75	89971
LA50	154
LGA50	161
LA75	359
LGA75	383

Abbreviations: bp: base pairs; #: Number of

B.1.1.3 Bowtie2 mapping results

Output from Bowtie2 for the alignment to the *de novo* assembly:

```
39715449 reads; of these:
  39715449 (100.00%) were paired; of these:
    10315711 (25.97%) aligned concordantly 0 times
    29018333 (73.07%) aligned concordantly exactly 1 time
    381405 (0.96%) aligned concordantly >1 times
    ----
    10315711 pairs aligned concordantly 0 times; of these:
      9374430 (90.88%) aligned discordantly 1 time
    ----
    941281 pairs aligned 0 times concordantly or discordantly; of these:
      1882562 mates make up the pairs; of these:
        471288 (25.03%) aligned 0 times
        501547 (26.64%) aligned exactly 1 time
        909727 (48.32%) aligned >1 times
99.41% overall alignment rate
```

Output from Bowtie2 for the alignment to the control genome:

```
39715449 reads; of these:
  39715449 (100.00%) were paired; of these:
    9744216 (24.54%) aligned concordantly 0 times
    28669492 (72.19%) aligned concordantly exactly 1 time
    1301741 (3.28%) aligned concordantly >1 times
    ----
    9744216 pairs aligned concordantly 0 times; of these:
      9130270 (93.70%) aligned discordantly 1 time
    ----
    613946 pairs aligned 0 times concordantly or discordantly; of these:
      1227892 mates make up the pairs; of these:
        458 (0.04%) aligned 0 times
        128079 (10.43%) aligned exactly 1 time
        1099355 (89.53%) aligned >1 times
100.00% overall alignment rate
```

B.1.1.4 Pipeline usage in the *de novo* assembly experiment with simulated reads from *Arabidopsis thaliana* to test for read mismapping

Command usage: `java fps.RegionQuantifier <List of SNPs file> <FASTA file> <Truncate name at space character? true | false> <BAM file> <Read length of the simulated dataset> <outputFile> <Run BLAST against any specific database? true | false> <Target BLAST database file | none>`

where:

`<List of SNPs file>` The absolute path of the text file with the list of heterozygous SNPs to be analysed by the pipeline.

`<FASTA file>` The absolute path of the Velvet assembly file.

`<Truncate name at space character? true | false>`

`true` should be chosen if the contigs' names have spaces.

`<BAM file>` The absolute path of the alignment/mapping file.

`<Read length of the simulated dataset>`

The read length (in base pairs) of the simulated read dataset. Used for internal computation by the pipeline. In this case, chosen as 150.

`<outputFile>` The name to be assigned for the output file.

`<Run BLAST against any specific database? true | false>`

`true` should be chosen, in this case, to allow the retrieval of the original genomic location correspondent to the SNP site being evaluated by the pipeline.
`<false>` should NOT be chosen in the current development stage of the tool.

`<Target BLAST database file | none>`

The absolute path of the BLAST database file.
`<none>` should NOT be chosen in the current development stage of the tool.

To speed up the computation for the control approach, the *A. thaliana* five chromosomes

had their start and end coordinates hard in the code of a similar pipeline. Its usage is shown below:

Command usage: `java fps.RegionQuantifierCONTROLS <List of SNPs file> <FASTA file> <Truncate name at space character? true | false> <BAM file> <Read length of the simulated dataset> <outputFile> <Run BLAST against any specific database? true | false> <Target BLAST database file | none>`

where:

`<List of SNPs file>`	The absolute path of the text file with the list of heterozygous SNPs to be analysed by the pipeline.
`<FASTA file>`	The absolute path of the control genome file.
`<Truncate name at space character? true | false>`	`false` was used in this case.
`<BAM file>`	The absolute path of the alignment/mapping file.
`<Read length of the simulated dataset>`	The read length (in base pairs) of the simulated read dataset. Used for internal computation by the pipeline. In this case, chosen as 150.
`<outputFile>`	The name to be assigned for the output file.
`<Run BLAST against any specific database? true | false>`	

`true` should be chosen, in this case, to allow the retrieval of the original genomic location correspondent to the SNP site being evaluated by the pipeline. `<false>` should NOT be chosen in the current development stage of the tool.

`<Target BLAST database file | none>`

The absolute path of the BLAST database file correspondent to the control genome. `<none>` should NOT be chosen in the current development stage of the tool.

B.1.2 FP SNP sites genomic locations tables

Tables comprising the FP SNP sites genomic locations detected by the pipeline runs over the alignments to the *de novo* assembly and control genome reference sequences are available at: https://github.com/acbellorib/fpSNPsProject.

B.1.3 Software availability

The source code used in this specific study is available at: https://github.com/acbellorib/fpSNPsProject.

Appendix C

C.1 A multifactorial experiment to evaluate false positive SNP generation due to read mismapping – supplementary information

C.1.1 Read simulation – additional information

C.1.1.1 SimSeq configuration planning

The Allpaths-LG manual (revision of 27-Jan-13 2:47:00 PM), section "Supported library constructions", and the work of Earl et al. (2011) were used as guidelines for setting up the parameter values for the read simulation stage. The calculations used for determining the parameters are summarised in the Tables C.1, C.2, C.3, and C.4.

Table C.1: Original genome sequence statistics. Adapted from Ribeiro et al. (2015).

Chromosome	Number of base pairs (bp)
1	30,427,671
2	19,698,289
3	23,459,830
4	18,585,056
5	26,975,502
Total number of bases	119,146,348

Table C.2: Coverage depth calculations. Adapted from Ribeiro et al. (2015).

Coverage depth	Number of base pairs (bp)
50-fold	5,957,317,400
100-fold	11,914,634,800
Total number of bases	119,146,348

Table C.3: Number of reads per read length dataset. Adapted from Ribeiro et al. (2015).

Read length dataset (bp)	# reads required (fragment library with 100-fold coverage depth)	# PE reads per FASTQ file (raw)	# PE reads per FASTQ file (rounded)
50	238,292,696	119,146,348	119,146,348
100	119,146,348	59,573,174	59,573,174
150	79,430,898.67	39,715,449.33	39,715,449
300	39,715,449.33	19,857,724.67	19,857,725
500	23,829,269.60	11,914,634.80	11,914,635
1,000	11,914,634.80	5,957,317.40	5,957,317
Read length dataset (bp)	# reads required (jumping library with 50-fold coverage depth)	# PE reads per FASTQ file (raw)	# PE reads per FASTQ file (rounded)
150	39,715,449.33	19,857,724.67	19,857,725

Abbreviations: bp: base pairs; #: Number of; PE: Paired-end

Table C.4: Insert lengths and standard deviations for each read length dataset. Adapted from Ribeiro et al. (2015).

Read length dataset (bp)	Fragment library short insert length (bp)	Standard deviation (bp)
50	90	9
100	180	18
150	270	27
300	540	54
500	900	90
1,000	1,800	180
Read length dataset (bp)	Jumping library short insert length (bp)	Standard deviation (bp)
150	3,000	300

Abbreviation: bp: base pairs

C.1.1.2 SimSeq configuration

– SimSeq command usage as described at St. John (2014):

Usage: `java -jar -Xmx2048m SimSeq.jar [required options] [options]`

Usage example:
```
java -jar Xmx10g SimSeq.jar -1 50 -2 50 --insert_size
90 --insert_stdev 9 --read_number 119146348 --read_prefix
50bp_AT_SimSeq_1st_PE --reference TAIR10_All5Chrms.fasta
--inf_id --out 50bp_AT_SimSeq_1st_PE.sam
```

Usage example:
```
java -jar Xmx10g SimSeq.jar -1 150 -2 150 --insert_size
 3000 --insert_stdev 300 --mate_pair --mate_frag 500
--mate_frag_stdev 50 -mate_pulldown_error_p 0.0
--read_number 19857725 --read_prefix 150bp_AT_SimSeq_1st_MP
--reference TAIR10_All5Chrms.fasta --inf_id
```

```
--out 150bp_AT_SimSeq_1st_MP.sam
```

where:

`-1 <argument>`	Integer length of first read, accordingly to the aimed read length dataset in this case.
`-2 <argument>`	Integer length of second read, accordingly to the aimed read length dataset in this case.
`--insert_size <argument>`	Mean library insert size for either mate-paired or paired-end reads, accordingly to the aimed read length in this case.
`--insert_stdev <argument>`	Mean library insert standard deviation for either mate-paired or paired-end reads, accordingly to the aimed read length.
`--read_number <argument>`	Integer number of reads to be sampled, in order to provide 100-fold coverage depth, accordingly to the aimed read length.
`--read_prefix <argument>`	Prefix for the simulated reads (e.g. `50bp_AT_SimSeq_1st_PE`).
`--reference <argument>`	Reference genome sequence file in uncompressed FASTA format. (REQUIRED)
`--inf_id`	Flag to output location information in the read identifier.

`--out <argument>` Filename for output SAM file. (REQUIRED)

Jumping library dataset specific options:

`--mate_pair` Flag to perform mate-pair rather than paired-end run.

`--mate_frag_stdev <argument>` Loop fragmentation standard deviation, set up with the default value of `50`.

`--mate_pulldown_error_p <argument>` Probability that a read does not include the biotin marker, set up as `0.0` in this case.

`--read_number <argument>` Integer number of reads to be sampled in order to provide 50-fold coverage depth for the aimed 150 bp read length dataset in this case.

In order to emulate a Phred quality score of 40 (considering Phred+33 scale) evenly for all the bases across the read length datasets, the base quality scores in the intermediate SAM files produced within the SimSeq pipeline had the "~" character substituted with an "`I`", using the Unix sed command. After these substitutions, the SimSeq usage example, available at St. John (2014), served as a guideline for the remaining steps of the pipeline for the production of the simulated reads, in FASTQ format, corresponding to each read length dataset. The steps performed were as follows:

– Creation of the "`.size`" file for the SAM to BAM format conversion:

Usage: `faSize [command flags] file(s).fa`

Usage example: `./SimSeq-master/cUtils/faSize -detailed -tab <in.fa>`

`> <out.size>`

`-detailed`	Flag to output name and size of each record.
`-tab`	Flag to output statistics in a tab separated format.
`<in.fa>`	Original sequences input file in FASTA format.
`<out.size>`	Desired output file labelled with `.size` extension in this case.

– SAM to BAM format conversion:

Usage: `samtools view [options] <in.bam>|<in.sam> [region1 []]`

Usage example: `samtools view -bS T <in.reference> -t <in.size>`

`-o <out.bam> <in.sam>`

`-b`	Flag to output BAM.
`-S`	Flag to specify that input is SAM.
`-T <in.reference>`	Reference sequence file.
`-t <in.size>`	List of reference names and lengths (`.size` file in this case).
`-o <out.bam>`	Output `.bam` filename.
`<in.sam>`	Input `.sam` file.

– BAM file sorting:

Usage: `samtools sort [options] <in.bam> <out.prefix>`

Usage example: `samtools sort <in.bam> <out.sorted>`

`<in.bam>`	Input `.bam` file.
`<out.sorted>`	Output sorted `.bam` filename.

– Sorted BAM file indexing:

Usage: `samtools index <in.bam> [out.index]`

Usage example: `samtools index <in.sorted.bam>`

`<in.sorted.bam>`	Input sorted `.bam` file.

– SAM to FASTQ conversion:

Usage: `SamToFastq [options]`

Usage example: `java -jar -Xmx10g ./picard-tools-1.119/SamToFastq.jar INPUT=File FASTQ=File SECOND_END_FASTQ=File INCLUDE_NON_PF_READS=true VALIDATION_STRINGENCY=SILENT`

`INPUT=File`	Input sorted `.bam` file to extract reads from. (REQUIRED)
`FASTQ=File`	Output FASTQ file (single-end FASTQ or, if paired, first end of the pair FASTQ). (REQUIRED)
`SECOND_END_FASTQ=File`	Output FASTQ file (if paired, second end of the pair FASTQ).
`INCLUDE_NON_PF_READS=true`	Include non-PF reads from the SAM file into the output FASTQ files. PF means 'passes filtering'. Reads whose 'not passing quality controls' flag is set are non-PF reads.
`VALIDATION_STRINGENCY=SILENT`	Opted validation stringency for all SAM files read by this program.

C.1.2 *De novo* assembly – additional information

C.1.2.1 Velvet configuration

Velvet command usage was based on the manual version 1.1 and a specific recommendation about mate-paired sequences usage described at The Genome Factory (2012). Due to this, mate-paired reads were initially reverse-complemented using the EMBOSS revseq tool version EMBOSS:6.6.0.0 (Emboss, 1999):

– revseq step:

Usage: `revseq [-sequence] <argument> -sformat1 <argument> [-outseq] <argument> -osformat2 <argument> -[no]tag`

Usage example: `revseq -sequence 150bp_AT_SimSeq_1st_MP_1.fastq -sformat1 fastq-sanger -outseq rc_150bp_AT_SimSeq_1st_MP_1.fastq -osformat2 fastq-sanger notag`

Usage example: `revseq -sequence 150bp_AT_SimSeq_1st_MP_2.fastq -sformat1 fastq-sanger -outseq rc_150bp_AT_SimSeq_1st_MP_2.fastq -osformat2 fastq-sanger notag`

`[-sequence] <argument>`	(Gapped) nucleotide sequence(s) filename and optional format, reference (input USA). (REQUIRED)
`-sformat1 <argument>`	Input sequence format. In this case, fastq-sanger.
`[-outseq] <argument>`	`[<sequence>.<format>]` Sequence set(s) filename

	and optional format (output USA). (REQUIRED)
-osformat2 <argument>	Output sequence format. In this case, fastq-sanger.
-[no]tag	[Y] Set this to false if you do not wish to add 'Reversed:' to the sequence description.

– velveth step:

Usage: ./velveth directory hash_length {[-file_format][-read_type] [-separate|-interleaved] filename1 [filename2 ...]} {...} [options]

Usage example: velveth . 96 -shortPaired -separate -fastq 150bp_AT_SimSeq_1st_PE_1.fastq 150bp_AT_SimSeq_1st_PE_2.fastq -shortPaired2 -separate -fastq rc_150bp_AT_SimSeq_1st_MP_1.fastq rc_150bp_AT_SimSeq_1st_MP_2.fastq

directory	Directory name for output files.
hash_length	Odd integer (if even, it will be decremented) <= 99 (if above, will be reduced). Note: although the parameter was input as 96, in order to make it as comparable as possible with the fixed hash length used by the Allpaths-LG assembler the hash length value was automatically decremented to 95 by the velveth package.
[-file_format]	File format option. In this case, -fastq.

`[-read_type]` Read type option. In this case, `-shortPaired`.

`[-separate|-interleaved]` File layout options for paired reads (only for FASTA and FASTQ formats). In this case, `-separate`, meaning "read 2 separate files for paired reads".

`filename1` Path to sequence file in this case.

`[filename2 ...]` Path to second sequence file in this case.

– velvetg step:

Usage: `./velvetg directory [options]`

Usage example: `velvetg . -read_trkg yes -amos_file yes -exp_cov auto -cov_cutoff auto -shortMatePaired2 yes`

`directory` Working directory name.

`-read_trkg <yes|no>` Tracking of short read positions in assembly. In this case, `yes`.

`-amos_file <yes|no>` Export assembly to AMOS file. In this case, `yes`.

`-exp_cov <floating point|auto>` Expected coverage of unique regions or allow the system to infer it. In this case, `auto`.

`-cov_cutoff <floating-point|auto>` Removal of low coverage nodes AFTER tour bus or allow the system to infer it.

	In this case, `auto`.
`-shortMatePaired* <yes|no>`	For mate-pair libraries, indicates that the library might be contaminated with paired-end reads. In this case, `-shortMatePaired2 yes`. Note: Parameter included here just for the sake of consistency with the recommendation about usage of "mate-paired sequences" with Velvet.

C.1.2.2 Allpaths-LG configuration

Allpaths-LG command usage was based on the software package embedded manual revision of 27-Jan-13 2:47:00 PM:

– Input files examples obtained after performing the steps described in the manual's section "Preparing data for ALLPATHS":

in_groups.csv

```
group_name,library_name,file_name
1,150bp_AT_SimSeq_1st_PE,
/mnt/scratch/ar41690/1st_AT_SimSeq/150bp_AT_SimSeq_1st_PE_*.fastq
```

```
2,150bp_AT_SimSeq_1st_MP,
/mnt/scratch/ar41690/1st_AT_SimSeq/150bp_AT_SimSeq_1st_MP_*.fastq
```

in_libs.csv

```
library_name,project_name,organism_name,type,paired,frag_size,frag_stddev,
insert_size,insert_stddev,read_orientation,genomic_start,genomic_end
150bp_AT_SimSeq_1st_PE,AT_AllpathsLG_1stAssy,A.thaliana,fragment,1,270,27,
,,inward,,
150bp_AT_SimSeq_1st_MP,AT_AllpathsLG_1stAssy,A.thaliana,jumping,1,,,3000,
300,outward,,
```

ploidy

```
1
```

– Perl script "`PrepareAllPathsInputs.pl`" run:

Usage: `PrepareAllPathsInputs.pl`

```
DATA_DIR=<full_path to REFERENCE DIR>/mydata
PICARD_TOOLS_DIR=/opt/picard/bin
```

Usage example: `PrepareAllPathsInputs.pl`

```
DATA_DIR=/mnt/scratch/ar41690/1st_AT_SimSeq/AT_AllpathsLG_1stAssy/
REFERENCE/DATA/ PICARD_TOOLS_DIR=/opt/picard/bin
```

DATA_DIR=<full_path to REFERENCE DIR>/mydata	Target data directory.
PICARD_TOOLS_DIR=/opt/picard/bin	Path to Picard directory.

– "RunAllPathsLG" pipeline run:

Usage: RunAllPathsLG arg1=value1 arg2=value2 ...

Usage example: RunAllPathsLG PRE=/mnt/scratch/ar41690/1st_AT_SimSeq/ AT_AllpathsLG_1stAssy/ DATA_SUBDIR=DATA RUN=RUN REFERENCE_NAME=REFERENCE TARGETS=standard

PRE=<full_path to PRE DIR>	The root directory in which the ALLPATHS pipeline directory will be created.
DATA_SUBDIR=<DATA directory name>	The DATA (project) directory name.
RUN=<RUN directory name>	The RUN (assembly pre-processing) directory name.
REFERENCE_NAME=<REFERENCE directory name>	The REFERENCE (organism) directory name.
TARGETS=<value>	Determines the operations performed by the pipeline. In this case, standard.

Table C.5: QUAST results for each assembly replicate. Adapted from Ribeiro et al. (2015).

Assembly detail	First Velvet assembly	Second Velvet assembly	First Allpaths-LG assembly	Second Allpaths-LG assembly
# contigs (>= 0 bp)	4,251	4,252	607	586
# contigs (>= 1,000 bp)	403	401	607	586
Total length (>= 0 bp)	119,084,054	119,053,539	115,964,030	115,898,343
Total length (>= 1,000 bp)	117,876,165	117,840,546	115,964,030	115,898,343
# contigs	842	845	607	586
Largest contig	8,578,385	6,688,511	3,993,636	3,183,886
Total length	118,169,135	118,138,303	115,964,030	115,898,343
Reference length	119,146,348	119,146,348	119,146,348	119,146,348
GC(%)	36.01	36.01	35.99	35.99
Reference GC(%)	36.03	36.03	36.03	36.03
N50	2,924,849	2,808,495	901,890	844,369
NG50	2,855,571	2,808,495	883,049	838,397
N75	1,400,402	1,400,345	344,027	349,933
NG75	1,400,402	1,400,345	324,538	321,283
L50	11	13	37	37
LG50	12	13	39	39
L75	26	28	88	88
LG75	26	28	95	95
# misassemblies	132	145	183	191
# misassembled contigs	54	57	115	110
Misassembled contigs length	71,484,825	60,248,624	50,403,708	56,506,149
# local misassemblies	726	721	835	828
# unaligned contigs	0 + 46 part	0 + 35 part	0 + 1 part	0 + 0 part
Unaligned length	147,154	107,527	505	0
Genome fraction (%)	98.324	98.332	96.479	96.416
Duplication ratio	1.008	1.008	1.009	1.009

Continued on next page

Assembly detail	First Velvet assembly	Second Velvet assembly	First Allpaths-LG assembly	Second Allpaths-LG assembly
# N's per 100 kbp	574.51	537.53	665.70	654.05
# mismatches per 100 kbp	5.16	5.37	5.89	6.35
# indels per 100 kbp	0.42	0.42	1.17	1.23
# genes	28,186 + 113 part	28,185 + 111 part	27,967 + 267 part	27,970 + 263 part
Largest alignment	7,477,918	5,889,670	3,000,193	3,177,551
NA50	1,934,540	2,422,643	693,004	615,806
NGA50	1,934,540	2,422,643	663,650	609,210
NA75	1,025,291	988,328	268,585	255,409
NGA75	1,005,107	980,170	236,584	236,667
LA50	16	15	45	48
LGA50	16	15	48	50
LA75	36	34	111	121
LGA75	37	35	121	131

Abbreviations: bp: base pairs; #: Number of

C.1.3 Read mapping – additional information

To prevent technical problems in the downstream analysis, all read names were changed so the read identifiers would end in the suffixes "_R1" and "_R2" instead of the original "/1" and "/2" assigned by the read simulator. This was done using the Unix "sed" tool. Also, the Unix "tr" command and custom Java code were used to check for any occurrences of IUPAC ambiguity codes (Cornish-Bowden, 1985) in the read datasets and in the *de novo* and control assemblies, replacing them with "N" characters as necessary. Mappings were then performed using Bowtie2 and BWA-SW set up with the following parameters to map with the two mismatch stringency rates (2% and 14%) evaluated in the study:

C.1.3.1 BWA-SW configuration

The mapping stringency in BWA-SW is controlled by means of the minimum score threshold (-T parameter, see Li (2013)). Reads with a mapping score of less than -T will not be mapped. Based on observations from BWA-SW's SAM output, it was concluded that the maximum score is equal to the length of the read, and, for each mismatch, 5 units are deducted. The value for -T needs to be calculated separately for each read length.

– Example with a 100 bp read:

maximum score = 100

-T score threshold = 30

subtract threshold from maximum score: 100 - 30 = 70

divide difference by the penalty awarded for each mismatch: 70/5 = 14 mismatches

However, the actual computation of the alignment score does seem to vary slightly with read length. To establish actual values for -T, input data sets with 3 reads each for each read length and corresponding reference sequences (data not shown) were created. In each case, read 1 matched the reference perfectly, read 2 had the number of mismatches allowed based on a 2% mismatch rate, and read 3 had one more mismatch than allowed and should therefore be rejected. The actual cut-offs, based on the alignment scores observed in the SAM output, were slightly different from the theoretical values, as shown in Table C.6. The values for -T actually used for the experiment were those shown in the last column of each table.

Table C.6: Values for BWA-SW -T parameter (last column) – 2% mismatches. Adapted from Ribeiro et al. (2015).

Read length (bp)	# mismatches allowed value	max. score possible	max. -T allowed value (THEORETHICAL)	max. -T allowed value (ACTUAL)
50	1	50	45	45
100	2	100	90	90
150	3	150	135	135
300	6	300	270	275
500	10	500	450	460
1,000	20	1,000	900	920

Abbreviations: bp: base pairs; #: Number of; max.: maximum

Table C.7: Values for BWA-SW -T parameter (last column) – 14% mismatches. Adapted from Ribeiro et al. (2015).

Read length (bp)	# mismatches allowed	max. score value possible	max. -T allowed value (THEORETHICAL)	max. -T allowed value (ACTUAL)
50	7	50	15	23
100	14	100	30	45
150	21	150	45	65
300	42	300	90	132
500	70	500	150	220
1,000	140	1,000	300	440

Abbreviations: bp: base pairs; #: Number of; max.: maximum

– BWA-SW parameters used:

Usage: `bwa bwasw [options] <target.prefix> <query.fa> [query2.fa]`

Usage example: `bwa bwasw -T <int> -t <int> <in.db.fasta> <in.fq> <mate.fq> > <out.sam>`

`-T <int>`	Score threshold divided by a.
`-t <int>`	Number of threads.
`<in.db.fasta>`	Path to the file containing the reference assembly in FASTA format.
`<in.fq>`	Path to the file containing the first paired-end reads dataset in FASTQ format.
`<mate.fq>`	Path to the file containing the second paired-end reads dataset in FASTQ format.

`<out.sam>` Output `.sam` filename.

For BWA-SW SAM files, read group tags and headers were added with a custom script, as this was required by the GATK downstream analysis.

C.1.3.2 Bowtie2 configuration

The mismatch rate in Bowtie2 can be controlled with the `--score-min` parameter. From the Bowtie2 manual (Johns Hopkins University, 2014), item `--score-min <func>`: "This is a function of read length. For instance, specifying `L,0,-0.6` sets the minimum-score function `f to f(x) = 0 + -0.6 * x`, where `x` is the read length."

The first parameter ("L" in the example above) specifies a linear relationship between read length and the number of mismatches. The second parameter (0 in the example above) is the `y` intercept, and, the third parameter, the coefficient for the slope of the regression. To calculate the coefficient for the formula, the intended maximum mismatch score (obtained by multiplying the number of mismatches by the default penalty -6) is divided by the read length, like in this example which assumes a read length of 100 bp and a maximum number of two mismatches per read:

Coeff = (-6*2)/100 = -0.12

Thus, the score-min formula for Bowtie2, for the strict 2% mismatch rate used here, was

"L,0,-0.12", whereas the relaxed mismatch rate of 14% used a score-min formula of "L,0, -0.84" (=(-6*14)/100).

– Bowtie2 parameters used:

Usage: `bowtie2 [options]* -x <bt2-idx> {-1 <m1> -2 <m2> | -U <r>} [-S <sam>]`

Usage example: `bowtie2 -x <bt2-idx> -1 <m1> -2 <m2> -S <sam> --phred33 --score-min <func> -p <int> --rg-id <text> --rg <text>`

`-x <bt2-idx>`	Index filename prefix (minus trailing `.X.bt2`).
`-1 <m1>`	Files with #1 mates, paired with files in <m2>.
`-2 <m2>`	Files with #2 mates, paired with files in <m1>.
`-S <out.sam>`	File for SAM output.
`--phred33`	Qualities are Phred+33.
`--score-min <func>`	Minimum acceptable alignment score w/r/t read length.
`-p <int>`	Number of alignment threads to launch.
`--rg-id <text>`	Set read group id, reflected in `@RG` line and `RG:Z: opt` field.
`--rg <text>`	Add <text> ("lab:value") to `@RG` line of SAM header.

All BAM files were checked for multimapped reads and, where necessary, these were

filtered out via a custom script using SAMtools 0.1.18 (Li et al., 2012) and Picard Tools 1.119 (Broad Institute, 2014a). The latter was also used to, where necessary, filter out soft-clipped reads at the end of contigs. The range of percentages of discarded reads in the BAM files which had some sort of filtering applied was of 0.01% to 0.13%.

Additional information about the mappers and their parameters can be found at Li (2013) and Johns Hopkins University (2014).

C.1.4 SNP calling – additional information

C.1.4.1 FreeBayes configuration

The FreeBayes command line parameters used are described below:

Usage: `freebayes/0.9.18/bin/freebayes [OPTION] ... [BAM FILE] ...`

Usage example: `/mnt/apps/freebayes/0.9.18/bin/freebayes -b <in.bam> -f <in.reference> -v <out.vcf> -q <Q> -m <Q> -Z`

`-b <in.bam>`	Add `FILE` to the set of BAM files to be analysed.
`-f <in.reference>`	Use `FILE` as the reference sequence for analysis.
`-v <out.vcf>`	Output VCF-format results to `FILE`.
`-q <Q>`	Minimum base quality input filter. Exclude alleles from analysis if their supporting base quality is less than `Q`. In this study, this value has been always set as `10`.
`-m <Q>`	Minimum mapping quality input filter. Exclude alignments from

analysis if they have a mapping quality less than `Q`. In this study, this value has been set as `0` or `20`, depending on the combination of factors being evaluated.

`-Z` Include the reference allele in the analysis.

Additional information can be found at Garrison (2012) and in the FreeBayes embedded `-help` option.

C.1.4.2 GATK configuration

The parameters used are described below, for each of the components integrated in the pipeline script:

– Picard Tools MarkDuplicates step:

Usage: `MarkDuplicates [options]`

Usage example: `java -jar <pathToPicard>/MarkDuplicates.jar INPUT=<File> OUTPUT=<File> METRICS_FILE=<File> AS=<Boolean> MAX_FILE_HANDLES_FOR_READ_ENDS_MAP=<Integer> READ_NAME_REGEX=<String> VALIDATION_STRINGENCY=<String>`

`INPUT=<File>` One or more input SAM or BAM files to analyse.

`OUTPUT=<File>`	The output file to write marked records to. (REQUIRED)
`METRICS_FILE=<File>`	File to write duplication metrics to. (REQUIRED)
`AS=<Boolean>`	If `true`, assume that the input file is coordinate sorted even if the header says otherwise. In this study, this value has been always set to `true`.
`MAX_FILE_HANDLES_FOR_READ_ENDS_MAP=<Integer>`	Maximum number of file handles to keep open when spilling read ends to disk. In this study, this value has been always set to `1000`.
`READ_NAME_REGEX=<String>`	Regular expression that can be used to parse read names in the incoming SAM file. In this study, for BWA-SW related input BAM files, this value has been always set to `null` to suppress the functionality of the parameter.
`VALIDATION_STRINGENCY=<String>`	Validation stringency for all SAM files read by this program. In this study, for BWA-SW related input BAM files, this value has been always set to `LENIENT` to allow the processing to clear the MarkDuplicates stage.

– SAMtools BAM file indexing step:

Usage: `samtools index <in.bam> [out.index]`

Usage example: `samtools index <in.deduped.bam>`

`<in.deduped.bam>`	The output file produced by the MarkDuplicates step.

– GATKs realignment step 1 — target interval list generation:

Usage example: `java -jar <pathToGATK>/GenomeAnalysisTK.jar -T RealignerTargetCreator -R <reference_sequence> -I <input_file> -o <out> -nt <num_threads>`

`-R <reference_sequence>`	Reference sequence file.
`-I <input_file>`	Input file containing sequence data. In this case, the output BAM file produced by the MarkDuplicates step.
`-o <out>`	An output file created by the walker. In this case, the `target_intervals.list` file to be created in this step.
`-nt <num_threads>`	Number of data threads to allocate to this analysis.

– GATK realignment step 2 — Indel realignment:

Usage example: `java -jar <pathToGATK>/GenomeAnalysisTK.jar -T IndelRealigner -R <reference_sequence> -I <input_file> -targetIntervals <targetIntervals> -o $uniquePrefix.realigned.bam`

`-R <reference_sequence>`	Reference sequence file.
`-I <input_file>`	Input file containing sequence data. In this case, the output BAM file produced by the MarkDuplicates step.
`-targetIntervals <targetIntervals>`	Intervals file output from RealignerTargetCreator step.
`-o <out>`	Output BAM file. In this case, the realigned BAM file to be created in this step. `$uniquePrefix`, here, is a variable which corresponds to the name of a given factor combination run.

– GATK HaplotypeCaller step:

Usage example: `java -jar <pathToGATK>/GenomeAnalysisTK.jar`

```
-T HaplotypeCaller -R <reference_sequence> -I <input_file>
-o <out> -nct <num_cpu_threads_per_data_thread>
-mmq <min_mapping_quality_score>
```

`-R <reference_sequence>`	Reference sequence file.
`-I <input_file>`	Input file containing sequence data. In this case, the realigned BAM file produced by the IndelRealigner step.
`-o <out>`	File to which variants should be written.
`-nct <num_cpu_threads_per_data_thread>`	Number of CPU threads to allocate per data thread.
`-mmq <min_mapping_quality_score>`	Minimum read mapping quality required to consider a read for analysis with the HaplotypeCaller. In the study, this value has been set as `0` or `20`, depending on the combination of factors being evaluated.

Additional information can be found at Garrison (2012), Broad Institute (2012a), Broad Institute (2014a), and in the embedded `-h` option of each tool.

C.1.5 Pipeline usage in the multifactorial experiment to evaluate the FP SNP generation due to read mismapping

Command usage: `java fps.RegionQuantifier <List of SNPs file> <FASTA file> <Truncate name at space character? true | false> <BAM file> <Read length of the simulated dataset> <outputFile> <Run BLAST against any specific database? true | false> <Target BLAST database file | none>`

where:

`<List of SNPs file>`	The absolute path of the text file with the filtered list of bi-allelic SNPs with quality scores equal or higher than 20 to be analysed by the pipeline.
`<FASTA file>`	The absolute path of the Velvet assembly file.
`<Truncate name at space character? true | false>`	`true` should be chosen if the contigs' names have spaces.
`<BAM file>`	The absolute path of the alignment/mapping file.
`<Read length of the simulated dataset>`	The read length (in base pairs) of the simulated read dataset. Used for internal computation by the pipeline.
`<outputFile>`	The name to be assigned for the output file.
`<Run BLAST against any specific database? true | false>`	`true` should be chosen, in this case, to allow the

retrieval of the original genomic location correspondent to the SNP site being evaluated by the pipeline. <false> should NOT be chosen in the current development stage of the tool.

<Target BLAST database file | none>

The absolute path of the BLAST database file. <none> should NOT be chosen in the current development stage of the tool.

To speed up the computation for the control approach, the *A. thaliana* five chromosomes had their start and end coordinates hard in the code of a similar pipeline. Its usage is shown below:

Command usage: `java fps.RegionQuantifierCONTROLS <List of SNPs file> <FASTA file> <Truncate name at space character? true | false> <BAM file> <Read length of the simulated dataset> <outputFile> <Run BLAST against any specific database? true | false> <Target BLAST database file | none>`

where:

<List of SNPs file> The absolute path of the text file with the filtered list of bi-allelic SNPs with quality scores equal or

higher than 20 to be analysed by the pipeline.

`<FASTA file>` The absolute path of the control genome file.

`<Truncate name at space character? true | false>`

`false` was used in this case.

`<BAM file>` The absolute path of the alignment/mapping file.

`<Read length of the simulated dataset>`

The read length (in base pairs) of the simulated read dataset.

Used for internal computation by the pipeline.

`<outputFile>` The name to be assigned for the output file.

`<Run BLAST against any specific database? true | false>`

`true` should be chosen, in this case, to allow the retrieval of the original genomic location correspondent to the SNP site being evaluated by the pipeline. `<false>` should NOT be chosen in the current development stage of the tool.

`<Target BLAST database file | none>`

The absolute path of the BLAST database file correspondent to the control genome. `<none>` should NOT be chosen in the current development stage of the tool.

C.1.6 SNP manifests extraction

The following steps were executed to extract the SNP manifests:

(1) All SNP entries passing the depth filter that belonged to a given assembly replicate (and considering any combination of factors and mapping/SNP calling scenario), were combined to form a single VCF file.

(2) The UNIX commands "cat" and "sort" were used to sort this VCF file, so the vcfuniq executable (Garrison, 2013) could be run on it.

(3) The resulting non-redundant VCF file was then processed with custom Java code, so the corresponding SNP manifests could be extracted from the assembly file.

The same procedure was performed for the 'control' dataset.

C.1.7 SNP annotation detailed results

After performing the BLAST of the SNP manifests from each assembly replicate and the control reference sequence, the Unix "awk" tool was applied to remove redundant lines of each corresponding BLAST result file, based on the "qseqid" field. Using Microsoft Excel version 14.0.7149.5000, a list of annotation terms was searched for and quantified from the "salltitles" field. A similar approach was used to categorise and quantify the same terms from the available *A. thaliana* annotation. These were then compared and the following table and charts contain the results:

Table C.8: *Arabidopsis thaliana* annotation general composition *versus* unique SNP manifests. Adapted from Ribeiro et al. (2015).

Number of occurrences retrieved by the used annotation approach						
Categories defined for SNP characterisation	BLAST database	Allpaths-LG first run	Allpaths-LG second run	Velvet first run	Velvet second run	Controls (compiled)
family	10,530	5,030	5,053	2,427	2,523	1,317
intergenic	31,342	57,798	57,516	49,018	48,478	26,494
other CDS	13,115	5,347	5,137	2,368	2,269	993
pseudogene	876	1,345	1,044	523	538	346
repeat	1,410	407	611	303	345	131
reverse transcriptase	24	29	20	18	18	15
specific transposon / retrotransposon	20	–	–	–	–	–
transposable element gene	3,900	33,751	35,064	22,795	22,831	9,578
transposase	14	41	61	44	33	–
unknown protein	3,713	1,314	1,306	1,173	1,242	890
Totals	64,944	105,062	105,812	78,669	78,277	39,764

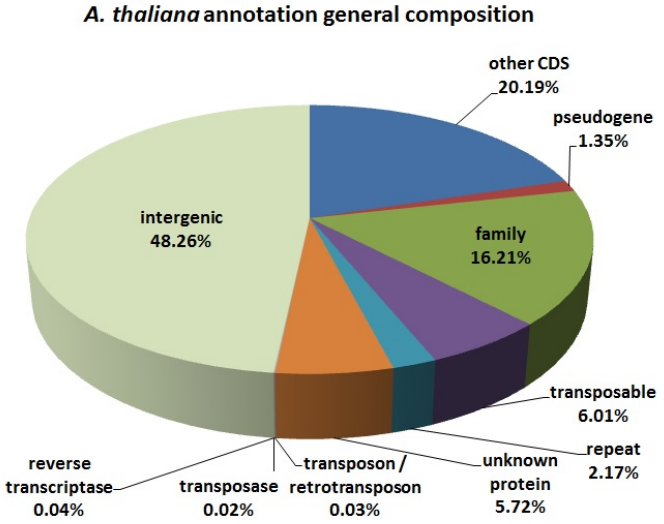

Figure C.1: General composition of the *Arabidopsis thaliana* annotation, as already referred to in Results section of Chapter 4. Adapted from Ribeiro et al. (2015).

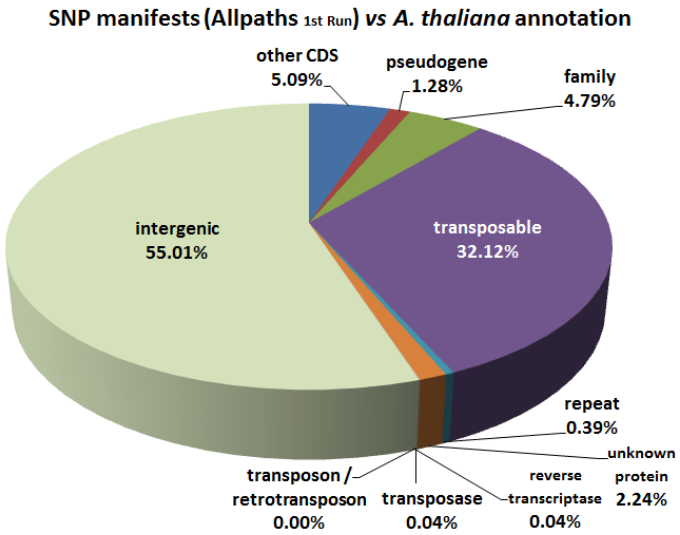

Figure C.2: BLAST-based annotation results for the SNP manifests from the Allpaths-LG first replicate of the experiment, as already referred to in Results section of Chapter 4. Adapted from Ribeiro et al. (2015).

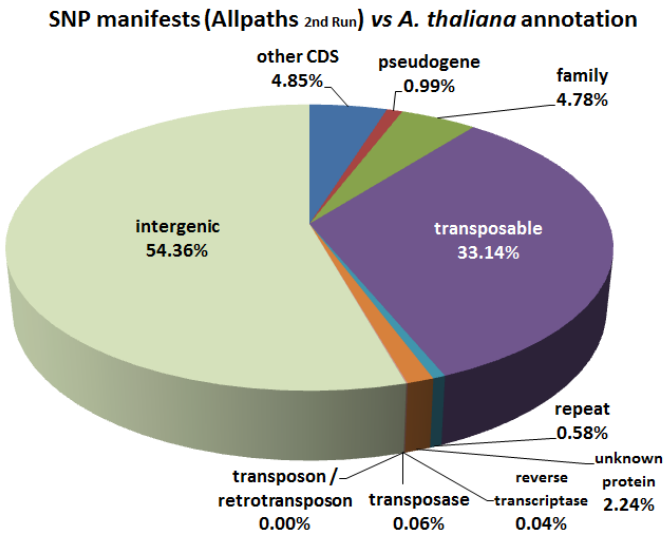

Figure C.3: BLAST-based annotation results for the SNP manifests from the Allpaths-LG second replicate of the experiment. Adapted from Ribeiro et al. (2015).

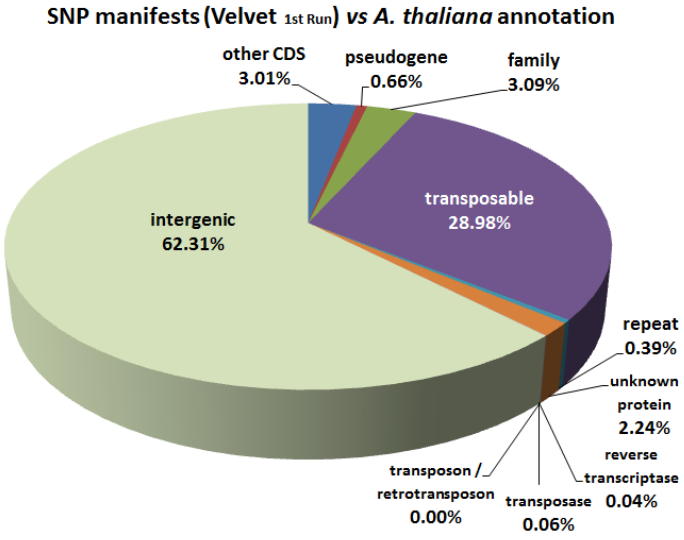

Figure C.4: BLAST-based annotation results for the SNP manifests from the Velvet first replicate of the experiment, as already referred to in Results section of Chapter 4. Adapted from Ribeiro et al. (2015).

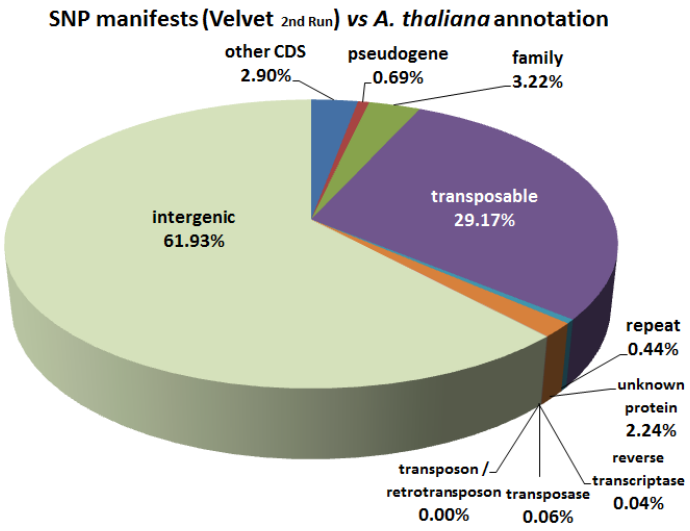

Figure C.5: BLAST-based annotation results for the SNP manifests from the Velvet second replicate of the experiment. Adapted from Ribeiro et al. (2015).

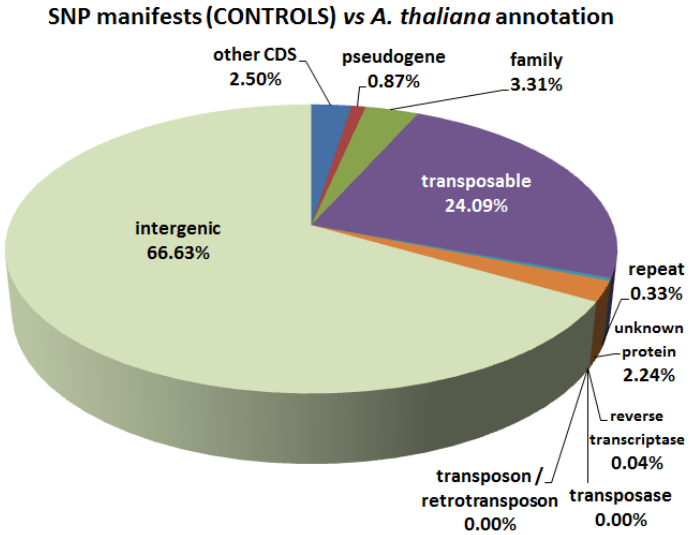

Figure C.6: BLAST-based annotation results for the SNP manifests from the two controls (compiled) of the experiment, as already referred to in Results section of Chapter 4. Adapted from Ribeiro et al. (2015).

C.1.8 FP SNP sites genomic locations by chromosomes

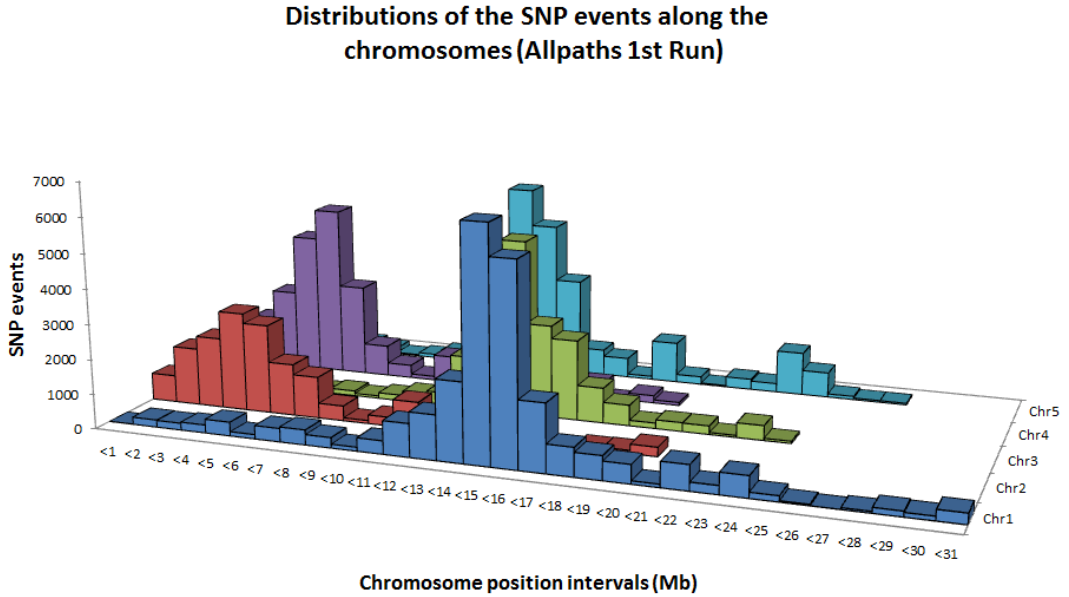

Figure C.7: FP SNP sites genomic locations (first Allpaths-LG *de novo* assembly replicate). Plot of the distributions of FP SNP sites, by chromosome, from the mapping to the first Allpaths-LG *de novo* assembly replicate. Genomic locations are shown on the x axis divided in intervals of up to 1 mega base pairs (only upper limits depicted for simplicity). FP SNP counts are shown on the y axis.

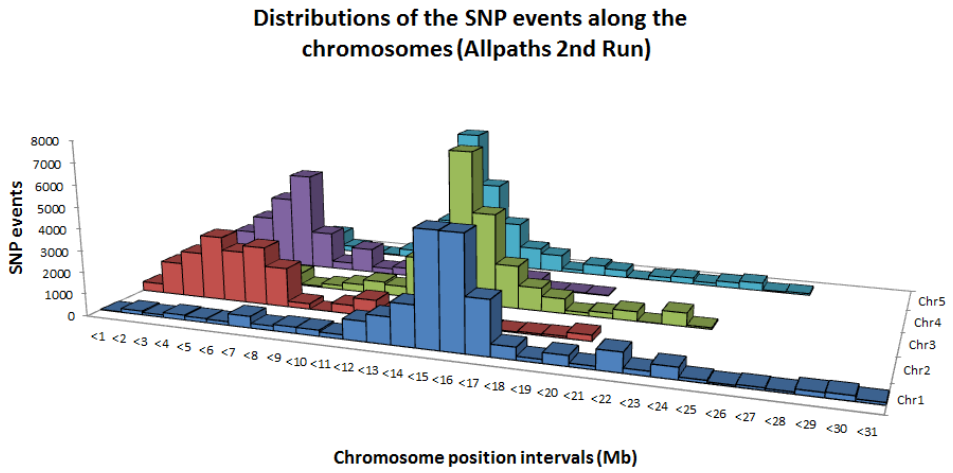

Figure C.8: FP SNP sites genomic locations (second Allpaths-LG *de novo* assembly replicate). Plot of the distributions of FP SNP sites, by chromosome, from the mapping to the second Allpaths-LG *de novo* assembly replicate. Genomic locations are shown on the x axis divided in intervals of up to 1 mega base pairs (only upper limits depicted for simplicity). FP SNP counts are shown on the y axis.

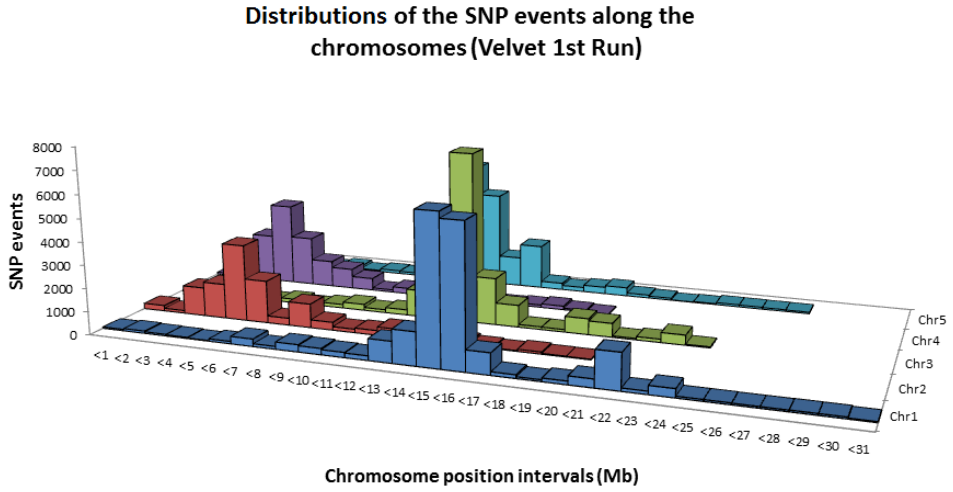

Figure C.9: FP SNP sites genomic locations (first Velvet *de novo* assembly replicate). Plot of the distributions of FP SNP sites, by chromosome, from the mapping to the first Velvet *de novo* assembly replicate (already referred to in Results section of Chapter 4). Genomic locations are shown on the x axis divided in intervals of up to 1 mega base pairs (only upper limits depicted for simplicity). FP SNP counts are shown on the y axis.

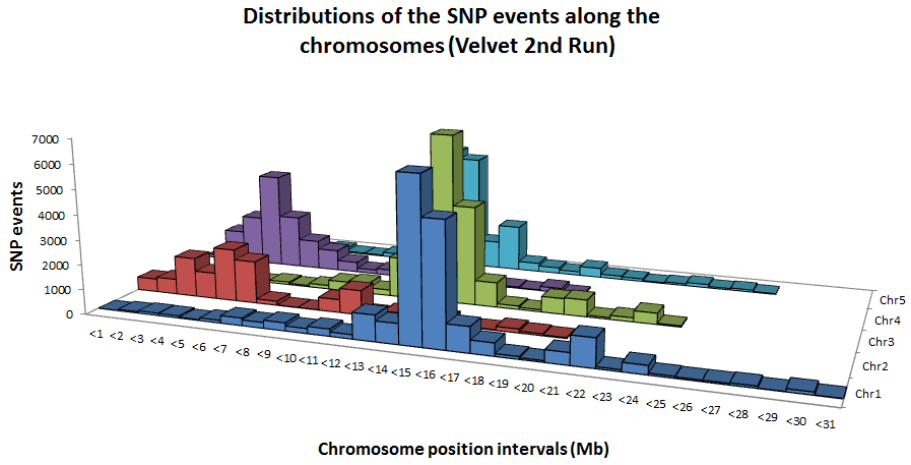

Figure C.10: FP SNP sites genomic locations (second Velvet *de novo* assembly replicate). Plot of the distributions of FP SNP sites, by chromosome, from the mapping to the second Velvet *de novo* assembly replicate. Genomic locations are shown on the x axis divided in intervals of up to 1 mega base pairs (only upper limits depicted for simplicity). FP SNP counts are shown on the y axis.

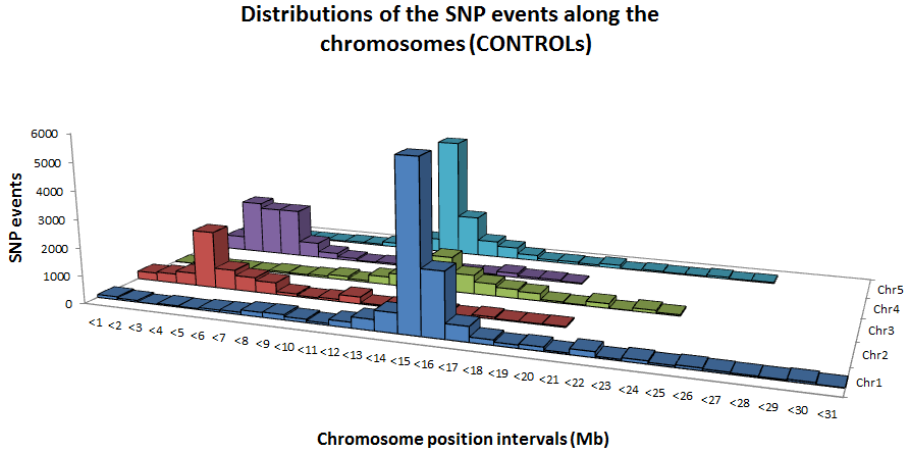

Figure C.11: FP SNP sites genomic locations from the compiled controls of the experiment. Plot of the distributions of FP SNP sites, by chromosome, from the mapping to the compiled controls of the experiment. Genomic locations are shown on the x axis divided in intervals of up to 1 mega base pairs (only upper limits depicted for simplicity). FP SNP counts are shown on the y axis.

C.1.9 Software availability

The direct download link for the file is:

http://ics.hutton.ac.uk/resources/antonio/fpSnpsCode.tar.gz.

The simulated reads used in the FP SNP study are available for download from the following URL: https://ics.hutton.ac.uk/resources/antonio/reads/.

C.1.10 Supplementary files

The supplementary spreadsheets mentioned in Chapter 4 ("ANOVA_Results.xlsx", "snpNumbersStats.xlsx", "readMappingStats.xlsx", and "avgPctOfMismapping.xlsx") can also be found at https://figshare.com/s/5da520bbd137fdfbfb89.

Appendix D

D.1 List of posters, presentations, and publications

D.1.1 International peer-refereed publication

Ribeiro A., Golicz A., Hackett C.A., Milne I., Stephen G., Marshall D., Flavell A., Bayer M. (2015) "An investigation of causes of false positive single nucleotide polymorphisms using simulated reads from a small eukaryote genome." BMC Bioinformatics 2015, 16:382 (11 November 2015). http://www.biomedcentral.com/1471-2105/16/382

D.1.2 Posters and presentations

Ribeiro A., Golicz A., Marshall D., Flavell A., Bayer M. (2013) "Exploring the origin of false positive SNPs in NGS data - Does reference sequence mis-assembly cause false positives ?". The James Hutton Institute annual postgraduate student competition, March 28th 2013, Aberdeen, UK. (poster)

Ribeiro A., Golicz A., Marshall D., Flavell A., Bayer M. (2013) "Does reference sequence mis-assembly cause false positive SNPs ?". COST (European Cooperation in Science and Technology) Action STATSeq Meeting, April 24th-26th 2013, Helsinki, Finland. (poster)

Ribeiro A., Golicz A., Marshall D., Flavell A., Bayer M. (2013) "Does reference sequence mis-assembly cause false positive SNPs ?". UK Genome Science Meeting 2013, September 2nd-4th 2013, University of Nottingham, Nottingham, UK. (poster)

Ribeiro A., Golicz A., Milne I., Marshall D., Flavell A., Bayer M. (2014) "The effect of NGS read length on the generation of false positive SNPs". The James Hutton Institute annual postgraduate student competition, March 20th 2014, Dundee, UK. (presentation)

Ribeiro A., Golicz A., Milne I., Stephen G., Marshall D., Flavell A., Bayer M. (2014) "The effect of read length on the generation of false positive SNPs in NGS data". COST (European Cooperation in Science and Technology) Action SeqAhead Scientific Meeting 'NGS after the Gold Rush', May 6th-7th 2014, The Genome Analysis Centre (TGAC), Norwich, UK. (poster)

Ribeiro A., Golicz A., Milne I., Stephen G., Marshall D., Flavell A., Bayer M. (2015) "Exploring the effect of read length on the generation of false positive SNPs in NGS data". The 39th NextGenBug Meeting, February 3rd 2015, The James Hutton Institute, Dundee, UK. (presentation)

Ribeiro A., Golicz A., Milne I., Stephen G., Marshall D., Flavell A., Bayer M. (2015) "Exploring the effect of read length on the generation of false positive SNPs in NGS data". Cell and Molecular Sciences Research Talks, February 20th 2015, The James Hutton Institute, Dundee, UK. (presentation)

Ribeiro A. (2015) "FP SNPs Busters: a quest to exterminate false positive SNPs from NGS data". The James Hutton Institute annual postgraduate student event, March 12th-13th 2015, Aberdeen, UK. (Macaulay Development Trust Sprent Prize **WINNER** presentation)

Ribeiro A., Golicz A., Hackett C.A., Milne I., Stephen G., Marshall D., Flavell A., Bayer M. (2015) "Exploring causes of false positive SNPs in NGS data". Division of Plant Sciences Seminar Series, June 10th 2015, The James Hutton Institute, Dundee, UK. (presentation)

Ribeiro A., Golicz A., Hackett C.A., Milne I., Stephen G., Marshall D., Flavell A., Bayer M. (2015) "Exploring causes of false positive SNPs in NGS data". Main Research Provider (MRP) inter-institute postgraduate student competition, June 23rd-24th 2015, Moredun Research Institute, Edinburgh, UK. (presentation)

Ribeiro A., Golicz A., Hackett C.A., Milne I., Stephen G., Marshall D., Flavell A., Bayer M. (2015) "Exploring causes of false positive SNPs in NGS data". The

41st NextGenBug Meeting, June 30th 2015, Centre for Genome-Enabled Biology and Medicine, University of Aberdeen, Aberdeen, UK. (presentation)